



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR3901T-W1

ACS 安全ブルートゥース® 接触式 カードリーダー

リファレンスマニュアル V1.02





改定履歴

リリース日付	改訂説明	バージョン
2017/08/30	<ul style="list-style-type: none">初回発布	1.00
2017/10/11	<ul style="list-style-type: none">セクション 5.1.2 更新: 電池寿命セクション 6.3 更新: 認証セクション 6.6 更新: 相互認証セクション 6.6.1 更新: SPH_to_RDR_ReqAuthセクション 6.6.3 更新: SPH_to_RDR_ReqAuthセクション 6.6.5 更新: SPH_to_RDR_DataReqセクション 6.6.6 更新: RDR_to_SPH_DataRspセクション 8.1.3 更新: ファームバージョン取得コマンドセクション 8.1.14 更新: デフォルト顧客マスターキーリセットリクエスト	1.01



リリース日付	改訂説明	バージョン
2019/10/28	<ul style="list-style-type: none">• セクション 6.0 の形式の更新 (ソフトウェア設計)• セクション 7.0 (ホストプログラミング API) の形式を更新する• セクション 6.1.3 の更新: 認証• アップデート 6.1.5: Bluetooth 通信プロトコル• アップデート 6.1.5.1: カードの電源を入れます• アップデート 6.1.5.2: カードの電源をオフにします• セクション 6.1.5.3 の更新: カードが存在するかどうかを確認します• セクション 6.1.5.4 の更新: APDU コマンド• セクション 6.1.5.5 を追加: APDU2 コマンド• セクション 6.1.5.6 の更新: ダイレクトコマンド• セクション 6.1.6.2 の更新: RDR_to_SPH_AuthRsp1• セクション 6.1.6.4 の更新: RDR_to_SPH_AuthRsp2• セクション 6.1.6.5 の更新: SPH_to_RDR_DataReq• 6.1.6.6 の更新: RDR_to_SPH_DataRsp	1.02
	<ul style="list-style-type: none">• 6.2.1.1 の更新: PC_to_RDR_IccPowerOn• セクション 6.2.1.4 の更新: PC_to_RDR_XfrBlock• 6.2.2.1 の更新: RDR_to_PC_DataBlock• セクション 6.2 の更新: USB 通信プロトコル• セクション 7.1.1 の更新: シリアル番号の取得コマンド• セクション 7.1.9 の更新: カードリセットシミュレーションオプションの設定• セクション 7.1.10 の更新: カードリセットシミュレーションオプションの取得• セクション 7.1.11 の更新: カードの応答時間間隔の設定• セクション 7.1.12 の更新: カードの応答時間間隔の取得• セクション 10.0 の削除: PC_to_RDR_XfrBlock を介して他のコマンドにアクセスします• 付録 A の更新: エラーコード• 付録 A: サポートされているカードタイプの削除• 製品プロモーション名を更新	

カタログ

1.0. 紹介	6
1.1. 参照ファイル.....	6
1.2. シンボルと略語.....	6
2.0. 特性	7
3.0. サポートしているスマートカード	8
3.1. MCU カード.....	8
3.2. メモリカード.....	8
4.0. システムのブロック図	9
5.0. ハードウェアのデザイン	10
5.1. 電池.....	10
5.1.1. 充電.....	10
5.1.2. 電池の寿命.....	10
5.2. Bluetooth インターフェース.....	10
5.3. USB インターフェース.....	11
5.3.1. 通信パラメーター.....	11
5.3.2. エンドポイント.....	11
5.4. ユーザーインターフェース.....	12
5.4.1. キー.....	12
5.4.2. LED ステータスインジケータ.....	12
5.5. スマートカードインターフェース.....	13
5.5.1. スマートカード電源 VCC (C1).....	13
5.5.2. プログラミング電圧 VPP (C6).....	13
5.5.3. カードタイプのセレクション.....	13
5.5.4. 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
5.5.5. カード引き裂き保護.....	13
6.0. ソフトデザイン	14
6.1. ブルートゥース通信プロトコル.....	14
6.1.1. Bluetooth の接続フロー.....	14
6.1.2. Profile セレクション.....	15
6.1.3. 認証.....	18
6.1.4. フレームフォーム.....	19
6.1.5. Bluetooth 通信プロトコル.....	20
6.1.6. 相互認証と暗号化プロトコル.....	33
6.2. USB 通信プロトコル.....	40
6.2.1. CCID Bulk-OUT メッセージ.....	42
6.2.2. CCID Bulk-IN メッセージ.....	47
7.0. ホストホストプログラミング API	50
7.1. 周辺機器制御 (Peripherals Control).....	50
7.1.1. シリアルナンバーを取得する (Get Serial Number).....	50
7.1.2. 乱数を取得する (Get Random Number) コマンド.....	51
7.1.3. ファームウェアのバージョンを入手する (Get Firmware Version).....	52
7.1.4. マスターキーを書き換える (Rewrite Master Key).....	53
7.1.5. スリープモードオプション (Sleep Mode Option).....	54
7.1.6. デバイスアドレスを取得する (Get Device Address).....	55
7.1.7. Tx パワーを設定する (Set Tx Power).....	56



7.1.8. Tx パワーを読み取る (Read Tx Power Value)	57
7.1.9. カードリセットシミュレーションオプション設定 (Set Card Reset Simulation Option)	58
7.1.10. カードリセットシミュレーションオプション取得 (Get Card Reset Simulation Option)	59
7.1.11. カードのレスポンスタイムインターバルの設定 (Set Card Response Time Interval)	59
7.1.12. カードのレスポンスタイムインターバルの読み取り (Get Card Response Time Interval)	61
7.1.13. ボタンをチェックする (Check button status)	62
7.1.14. 顧客マスターキーをリセットする請求 (Customer Master Key Reset Request)	63
7.2. メモリカードのコマンドセット	64
7.2.1. メモリカード – 1、2、4、8 および 16 kilobit I2C カード	64
7.2.2. メモリカード – 32、64、128、256、512 および 1024 kilobit I2C カード	66
7.2.3. メモリカード – ATMEL AT88SC153	68
7.2.4. メモリカード – ATMEL AT88C1608	72
7.2.5. メモリカード – SLE4418/SLE4428/SLE5518/SLE5528	75
7.2.6. メモリカード – SLE4432/SLE4442/SLE5532/SLE5542	79
7.2.7. メモリカード – SLE 4406/SLE 4436/SLE 5536/SLE 6636	84
7.2.8. メモリカード – SLE 4404	88
7.2.9. メモリカード – AT88SC101/AT88SC102/AT88SC1003	92
附录 A. エラーコード	99

図示カタログ

図 1 : ACR3901T-W1 アーキテクチャ	9
図 2 : Bluetooth 接続フロー	14
図 3 : nRFgo Studio GATT 配置ページ	16
図 4 : 認証手順	18

チャートカタログ

表 1 : シンボルと略語	6
表 2 : 推定される電池の寿命	10
表 3 : USB インターフェース配線	11
表 4 : LED ステータスインジケータ	12
表 5 : ACR3901T-W1 のサービスハンドルと UUID メッセージリスト	17
表 6 : ブルートゥースフレームフォーマット	19
表 7 : 相互認証した暗号化されたブルートゥースフレームフォーマット	20
表 8 : コマンドコードの概要	20
表 9 : コマンドコードの概要	21
表 10 : 相互認証コマンドの概要	34
表 11 : エラーコード	99

1.0. 紹介

ACR3901T-W1 安全 ACS Bluetooth®SIM サイズのスマートカードリーダーは、PC/モバイルデバイスとスマートカードリーダー間の通信インターフェースです。普通には異なるコマンドと通信プロトコルを使っているスマートカードはお互いに直接に通信できませんが、ACR3901T-W1 は様々のカードに統一のインターフェースを作れます。スマートカードの各項目を処理することで、多くの場合でスマートカードシステムの実装と関わりのない、業務の技術的な詳細に対し、コンピュータ・ソフトウェア・プログラマは責任を負う必要がありません。

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)
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

- USB フルスピード・インターフェース
- ブルートゥースインターフェース
- プラグアンドプレイ - CCID サポートにより、最大限の互換性と機動性を実現
- スマートカードリーダー:
 - ISO 7816 クラス A、B、C の (5V、3V および 1.8V) カードと SIM サイズカードをサポート
 - 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
 - 様々なメモ리카ードサポート
 - プロトコルとパラメータの選択サポート (PPS)
 - 短絡保護保有
 - AES-128 暗号化アルゴリズムサポート
- 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 の上のラッパを経由で)、CT- API をサポート
- 内蔵されている周辺機器:
 - LED
 - キー
- USB ファームウェアのアップグレード機能 ²
- Android™ 4.3 と以降のバージョンサポートしている ³
- iOS 5.0 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
 - ブルートゥース
 - PC/SC
 - CCID
 - CE
 - FCC
 - RoHS
 - REACH
 - Microsoft® WHQL

² PC リンクのモードに適用しています。

³ ACS の Android ライブラリを使用

⁴ ACS の iOS ライブラリを使用する

3.0. サポートしているスマートカード

3.1. MCU カード

ACR3901T-W1 スマートカードが PC/SC 仕様に準拠しているため、ISO 7816 A、B および C タイプ (5 V、3 V、1.8V) のスマートカードをサポートして、T=0 もしくは T=1 プロトコル準拠の全ての MCU カードもサポートしています。

カードの ATR が専用の操作モードを指定すれば (TA2 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 ACR3901T-W1 がこのモードをサポートできない場合、ACR3901T-W1 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、ACR3901T-W1 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、ACR3901T-W1 がその通信パラメータを使用して、PPS を実行する。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。

3.2. メモリカード

ACR3901T-W1 がサポートしているメモリカード、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
- 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 および AT88SC1608
- 書き込み保護機能付インテリジェント 1 キロバイトの 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
- '104' タイプ EEPROM (読み取りオンリー型トークンカウンタカード、以下を含めて:
 - Infineon®: SLE4406、SLE4436、SLE5536 および SLE6636
- インテリジェント 416 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®: SLE4404
- アプリケーションゾーンでのセキュリティロジックを使用したカード、以下を含めて:
 - Atmel®: AT88SC101、AT88SC102 および AT88SC1003

4.0. システムのブロック図

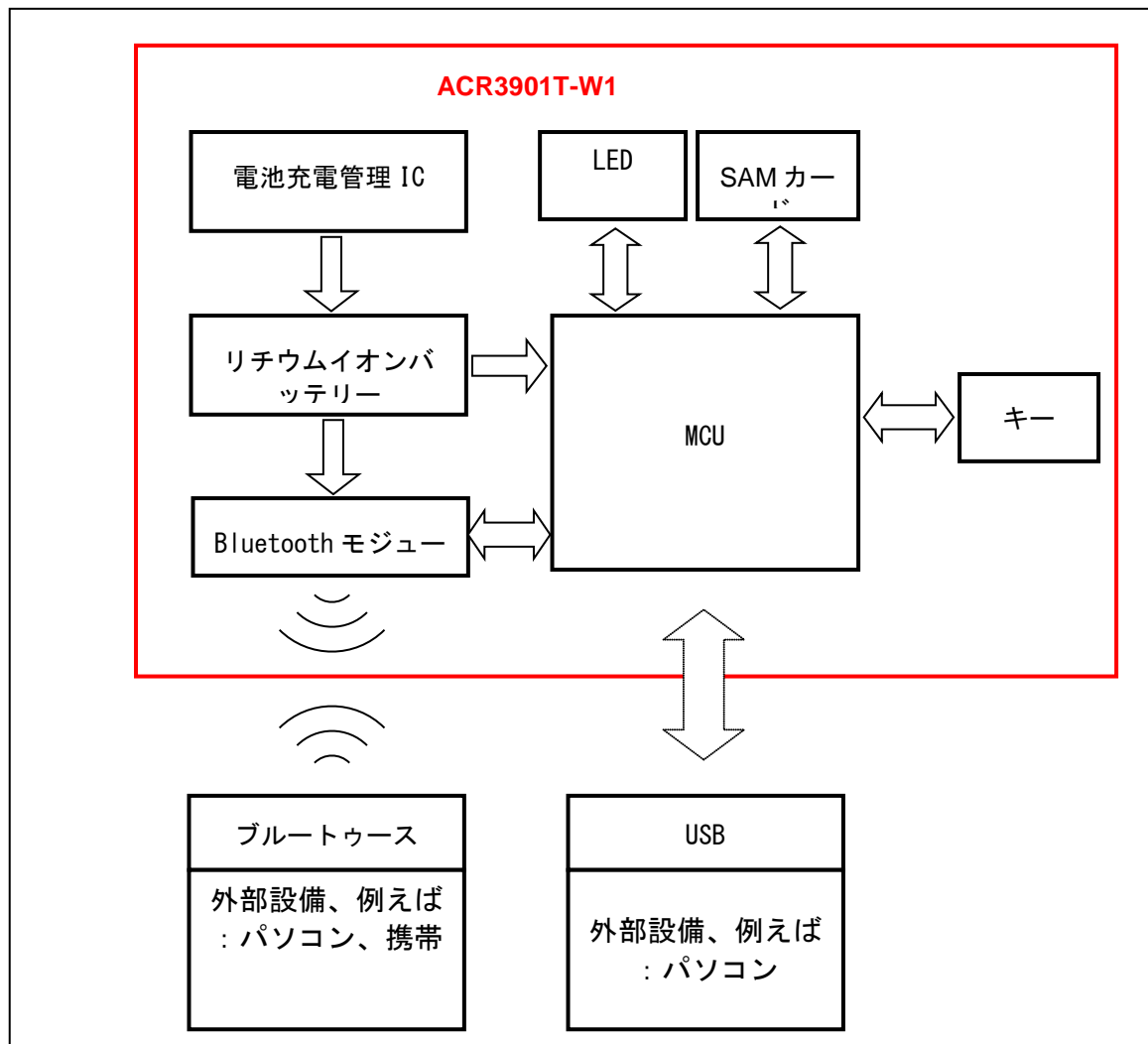


図1 :ACR3901T-W1 アーキテクチャ

5.0. ハードウェアのデザイン

5.1. 電池

ACR3901T-W1 は 70 ミリアンペアの充電式リチウムイオン電池を使用しています。

5.1.1. 充電

ACR3901T-W1 のバッテリーが切れると、次のいずれかのモードで充電することができる: OFF モード、USB モード、Bluetooth モード; 前提条件は電源コンセントに接続されています。

5.1.2. 電池の寿命

電池の駆動時間はデバイスの使用状況に依存します。モード条件に応じて、バッテリー寿命の推定値は以下のようになります。

モード	推定される電池の寿命
動作モード	4 日 ^{*(1)}
スタンバイモード	100 日 ^{*(2)}
オフモード	1 年

表2 : 推定される電池の寿命

***注:**異なるスマートカードで結果が違います。

⁽¹⁾ Bluetooth モードでは、1 分間の操作で 1 日に 10 回の操作を実行します。

⁽²⁾ Bluetooth モードでは、スリープ時間を 60 秒に設定し、1 日に 1 回ウェイクアップします。

5.2. Bluetooth インターフェース

ACR3901T-W1 は低消費のブルートゥース (LE) インターフェースで、デバイスを PC/モバイルデバイスとペアリングします。

5.3. USB インターフェース

Micro-USB が充電バッテリーのポートとして、コンピュータに ACR3901T-W1 を接続するために使用されています。このポートは、PC-リンクモードで ACR3901T-W1 を動作させるために使用されています。

5.3.1. 通信パラメーター

ACR3901T-W1 は USB 2.0 仕様の USB インターフェース介してコンピュータに接続されます。12 mbps の USB フルスピードモードで動作します。

ピン	信号	機能
1	V _{BUS}	カードに+5 V の電源を供給
2	D-	ACR3901T-W1 と PC は差動信号でデータを転送
3	D+	ACR3901T-W1 と PC は差動信号でデータを転送
4	GND	電源用の参照電圧レベル

表3 : USB インターフェース配線

5.3.2. エンドポイント

ACR3901T-W1 が下記のエンドポイントを介して、ホストと通信します：

エンドポイント制御 (Control Endpoint)	セットアップと制御
バルクアウト (Bulk OUT)	ホストから ACR3901T-W1 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)
バルクイン (Bulk IN)	ACR3901T-W1 からホストに返す応答に対して (データパケットサイズは 64 バイト)
割り込み入力 (Interrupt IN)	ACR3901T-W1 からホストに送信する状態メッセージに対して (データパケットサイズは 8 バイト)

5.4. ユーザーインターフェース

ACR3901T-W1 は二つのボタンと二つ LED インジケータを備えています。(単色: 赤; 二色: 緑とブルー)。

5.4.1. キー

ACR3901T-W1 は三つのモードを提供しています: USB、OFF とブルートゥース。ユーザーはデータ伝送インターフェースとして、一つのモードを選択することができます。

- ボタン(表) - ON、OFF またはリーダーを呼び覚ます; SAm カードを抜く操作を模擬します。
 - ボタンを押すと、リーダーが ON になって、デフォルトはブルートゥースモードです。パソコンと接続すると、USB モードに自動的に切り替えます。長く押すと設備が OFF になります。
- ボタン(裏) - リーダーをリセットする

5.4.2. LED ステータスインジケータ

違う操作状態を示すために、ACR3901T-W1 は二つの LED インジケータをついてます(単色: 赤; 二色: 緑とブルー)。

- 赤の LED - 電池状態
- ブルーの LED - ブルートゥースモードでカードとリーダーの状態
- 緑の LED - ブルートゥースモードでカードとリーダーの状態

色	LED ステータス	状態
赤	点灯	バッテリーは充電中です(バッテリーが完全に充電された後はオフになります)
	ゆっくり点滅 (1 秒/点滅)	充電必要
青	速い-ゆっくり点滅 (高速: 250ms /フラッシュ、 遅い: 500ms /フラッシュ)	Bluetooth デバイスとペアリング準備完了
	ゆっくり点滅 (2 秒/点滅)	ブルートゥース設備と接続済、カード操作なし
	速く点滅する	リーダーとモバイルデバイスがデータを通信している
	点灯	カードが接続して、電源いれました
緑	ゆっくり点滅 (2 秒/点滅)	カード操作なしで、ACR3901T-W1 は PC 命令を待っています
	速く点滅する	リーダーと PC がデータを通信している
	点灯	カードが接続して、電源いれました

表4 : LED ステータスインジケータ

注: 赤、青、緑の LED が同時に消灯の場合、ACR3901T-W1 の電源はオフになります。リーダーがブルートゥースモジュールからのエラーメッセージを受信すると、青と緑の LED が1秒で点灯して、消します。

5.5. スマートカードインターフェース

ACR3901T-W1 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、ACR3901T-W1 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する。

5.5.1. 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 よりも高くはない。

5.5.2. 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はないです。ACR3901T-W1 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。

5.5.3. 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ビする前に、ACR3901T-W1 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리카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。MCU カードは 1 つだけのプロトコル (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します。

5.5.4. 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。

5.5.5. 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、ACR3901T-W1 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、ACR3901T-W1 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ある。

注: ACR3901T-W1 は、決して挿入されたカードへの電源供給に切り替わりません。ホストから適切なコマンドが読者に送られ、明示的にこの操作を行う必要があります。

6.0. ソフトデザイン

6.1. ブルートゥース通信プロトコル

6.1.1. Bluetooth の接続フロー:

下記は Bluetooth の接続フローです:

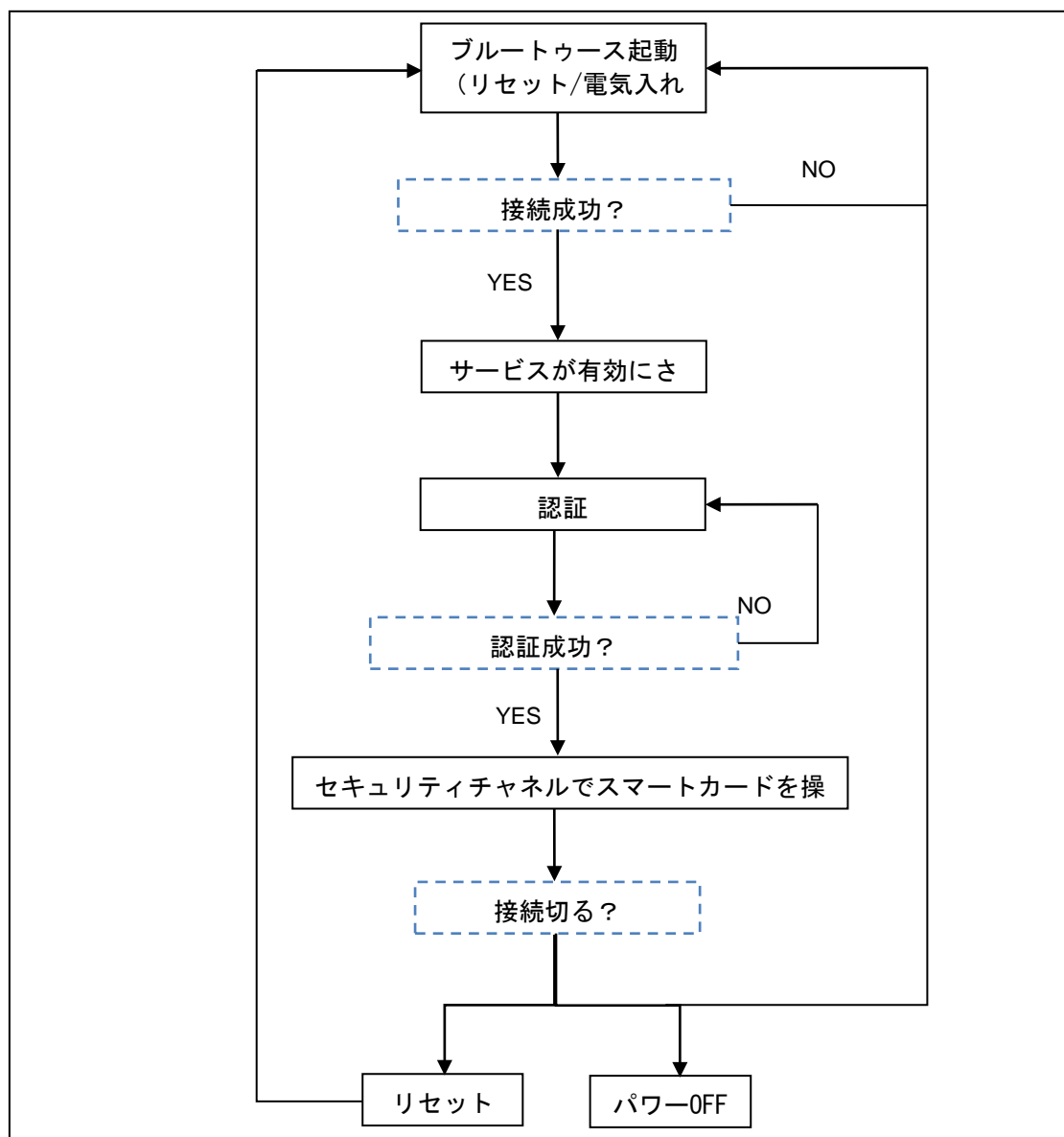


図2 :Bluetooth 接続フロー

6.1.2. Profile セレクション

ACR3901T-W1 はデータを送信するためのインターフェースとしての Bluetooth 技術を使用するように設計されたスマートカードリーダーです。3 つのパイプでコマンド通信できるカスタマイズされたサービスが使用されている：一つのパイプがコマンド要求のために使用され、二つのパイプは、コマンド応答に使用され、もう一つのパイプでカードのモードと Sleep モードの状態をペアリングのデバイスに知らせます。

また、リーダーは Bluetooth モードで動作している時、現在の電力消費量が重要なので、従って、カスタマイズのバッテリーサービスが現在のバッテリーステータスをペアリング装置に通知するために使用されます。電池状態の変化があった場合、リーダーは、特定のパイプを介してペアリング装置に通知します。簡単に言えば、電池の容量は三つのレベルに分けています。下記のテーブルは各レベルと対応の返す数値を示します：

状態	電圧	返ってきた数値
十分	$\geq 3.3 \text{ V}$	FEh
バッテリー不足	$< 3.3 \text{ V}$ 、また $\geq 2.9 \text{ V}$	FFh/FEh/00h 以外の数値
電源切れ	$< 2.9 \text{ V}$	00h
USB モード		FFh

カードの状態には変更があるまたはリーダーが Sleep モードに入る場合、カード状態通知サービスはペアリングサービスに知らせます。下記はステータステーブルと対応の返す数値：

状態	返ってきた数値
カード挿入されていない	50 02h
カードが挿入されました	50 03h
リーダーは Sleep モードに入りました	50 04h

最後に、ユーザに多くのリーダーの情報を提供するために、カスタマイズされたデバイス情報サービスが追加されます。これは、手動で読み、またはアプリケーションの要求によって読むことができます。特徴はモデル番号、シリアル番号、ファームウェアバージョン、およびメーカー名が含まれます。

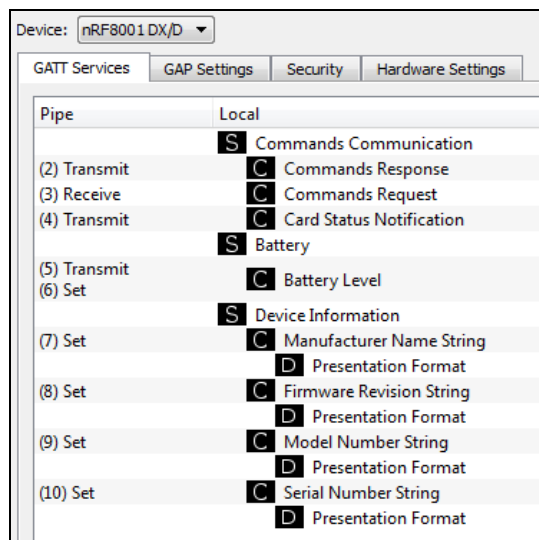


図3 : nRFGo Studio GATT 配置ページ

nRFGo-Studio 配置には一つのサービスが追加されて、総計 10 項のサービスがあります：

```
#define PIPE_GAP_DEVICE_NAME_SET 1
#define PIPE_COMMANDS_COMMUNICATION_COMMANDS_RESPONSE_TX 2
#define PIPE_COMMANDS_COMMUNICATION_COMMANDS_REQUEST_RX 3
#define PIPE_COMMANDS_COMMUNICATION_CARD_STATUS_NOTIFICATION_TX 4
#define PIPE_BATTERY_BATTERY_LEVEL_TX 5
#define PIPE_BATTERY_BATTERY_LEVEL_SET 6
#define PIPE_DEVICE_INFORMATION_MANUFACTURER_NAME_STRING_SET 7
#define PIPE_DEVICE_INFORMATION_FIRMWARE_REVISION_STRING_SET 8
#define PIPE_DEVICE_INFORMATION_MODEL_NUMBER_STRING_SET 9
#define PIPE_DEVICE_INFORMATION_SERIAL_NUMBER_STRING_SET 10
#define NUMBER_OF_PIPES 10
```

`#define PIPE_GAP_DEVICE_NAME_SET` が実行している間にもアプリケーションコントロールにデバイスネームを変更されます。ブルートゥースモードでは、公告ネームのフォーマットは“ACR3901T-W1XXXXXXX”です。その中“XXXXXXX”はリーダーのシリアルナンバーの最後の 7 バイトです。

公告ネームを“ACR3901T-W1XXXXXXX”にするために、ブルートゥースモードを有効にします。

ブルートゥースモードを有効にする：

1. 設定 (06h) 配置をブルートゥースモードにアップロードします。
2. チャンネル 1 でデバイスネームのフォーマットを“ACR3901T-W1XXXXXXX”に設定します。
(PIPE_GAP_DEVICE_NAME_SET)
3. 接続 (0Fh)。
4. 公告。

属性名称	UUID	ハンドル
------	------	------



属性名称	UUID	ハンドル
DeviceName	2A00	03h
Send (Reader → Paired device)	8002	0Bh
Receive (Paired device → Reader)	8003	0Eh
CardStatus	8004	10h
BatteryLevel	2A19	14h
Manufacturer	2A29	18h
FW_Version	2A26	1Bh
ModelNumber	2A24	1Eh
SerialNumber	2A25	21h

表5 : ACR3901T-W1 のサービスハンドルと UUID メッセージリスト

6.1.3. 認証

機密データが ACR3901T-W1 にロードする前に、データ処理サーバは、リーダ内部で保護されたデータを変更する権限のため ACR3901T-W1 によって認証されなければなりません。ACR3901T-W1 は相互認証を採用します。

良い説明のために、下の図示を参照してください(シンプルさとより良い説明のために、下の図示がブリッジデバイスを省略している)

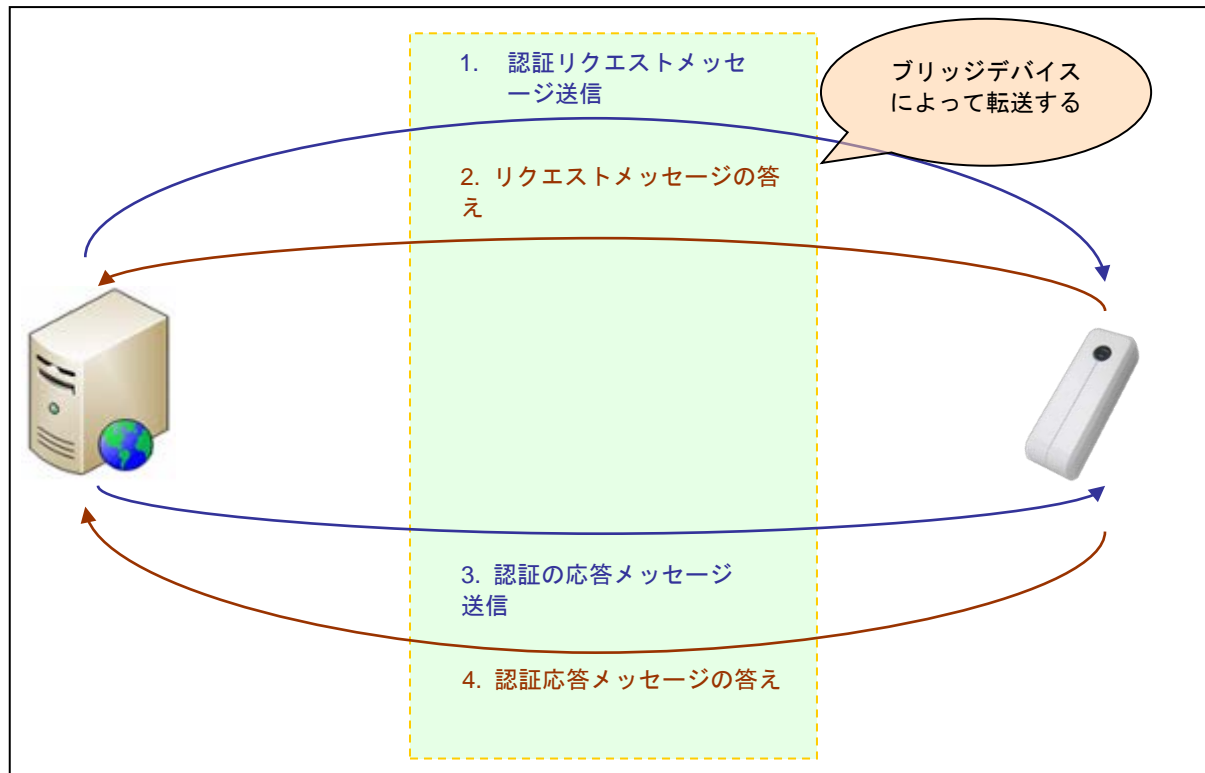


図4 : 認証手順

デフォルトの顧客マスターキー(16 進数): **FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF**

注意: 認証キーが誤って 6 回以上入力された場合、リーダーはロックされ、使用できません。。

関係情報は ACS の販売員までに連絡してください。

6.1.4. フレームフォーム

6.1.4.1. ブルートゥースフレームフォーマット

HID フレーム	長さ(バイト)	説明
識別子	1	コマンド
長さ	2	長さ {ペイロード+チェックサム}
ペイロード	0-N	データ
チェックサム	1	XOR {識別子、長さ、ペイロード}

表6 : ブルートゥースフレームフォーマット

フレームフォームは:

識別子 + LEN1 + LEN2 + N-バイトペイロード + チェックサム

コマンドの長さ(識別子、長さ及びペイロードを含めて)は20バイトより長い場合、リーダーまたはペアリングデバイスは自動的にいくつかのフレームに分けます。

データチェックサムは、無線にデータ送信時に導入された可能性のあるエラーを検出するのに使用されます。データチェックサムを計算する必要な場合: XOR {識別子、長さ、ペイロード}。

例: 62010063 => チェックサム = 63h

6.1.4.2. 相互認証したブルートゥースフレームフォーマット

誰かブルートゥース通信チャネルを介しての攻撃を防止するために、相互認証を投入します。相互認証が成功すると、表 6 中のブルートゥースフレームフォーマットが暗号化され、ヘッダ1バイト、Len2 バイト、チェック 1 バイトにカプセル化されます。相互認証したブルートゥースフレームフォーマットの構造は:

ヘッダ + Len + (識別子 + 長さ + ペイロード + チェックサム)* + チェックバイト

注: AES-128 CBC 暗号化モードを採用して、顧客マスターキーで各 16 バイトのデータを復号化します。AES-128 CBC 暗号化モードでは、初期ベクトルは 16 バイトです(00h)。

HID フレーム	長さ(バイト)	説明	
ヘッダバイト	1	値: 72h/22h	
Len	2	長さ {識別子 + 長さ + ペイロード + チェックサム + チェック+終止バイト}	
識別子	1	コマンド	ブルートゥースフレームフォーマットの暗号化データ: この部分の最終のデータの長さは 16*N バイト (N>0)
長さ	2	長さ {ペイロード+チェックサム}	
ペイロード	0-N	データ	
チェックサム	1	XOR {識別子、長さ、ペイロード}	
チェックバイト	1	XOR {ヘッダ、Len、暗号化された(識別子、長さ、ペイロード、チェックサム)}	

表7 : 相互認証した暗号化されたブルートゥースフレームフォーマット

6.1.5. Bluetooth 通信プロトコル

ACR3901T-W1 は、予め定義されたプロトコルを採用して、Bluetooth・インターフェースでペアリングされたデバイスと通信します。プロトコルは、CCID コマンドパイプと応答パイプの形式に似ています。

コマンド	サポートモード	送信側	説明
62h	認証済	ペアリングデバイス	ICC 電気入れる
63h	認証済	ペアリングデバイス	ICC 電源がオフになる (ICC Power Off)
65h	認証済	ペアリングデバイス	カードあるかどうかチェック (Get Card Presence)
6Fh	認証済	ペアリングデバイス	APDU 交換 (Exchange APDU)
61h	認証済	ペアリングデバイス	パラメーターを設定する
6Bh	認証済	ペアリングデバイス	外部コマンド
70h	接続済/認証済	ペアリングデバイス	SPH_to_RDR_ReqAuth*
71h	接続済/認証済	ペアリングデバイス	SPH_to_RDR_AuthRsp*

表8 : コマンドコードの概要

コマンド	サポートモード	送信側	説明
12h	認証済	リーダー	ICC Power On コマンドの応答
13h	認証済	リーダー	ICC Power Off コマンドの応答
14h	認証済	リーダー	Get Card Presence コマンドの応答
11h	認証済	リーダー	Exchange APDU コマンドの応答
...
16h	認証済	リーダー	設定のパラメーターに対して応答する
15h	認証済	リーダー	外部コマンドの応答
20h	接続済/認証済	リーダー	RDR_to_SPH_AuthRsp1*
21h	接続済/認証済	リーダー	RDR_to_SPH_AuthRsp2*

表9 : コマンドコードの概要

*注: これらのコマンド/応答は相互認証に使用される通信コードです。

6.1.5.1. カードに電源入れる (Card Power On)

このコマンドはリーダーにパワーアップのリクエストを送信するために使用されます。

コマンドのフォーマット

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>LEN1 LEN2 (wLength)</i>	2	0100h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>CSUM (wChecksum)</i>	1	63h	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット

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	12h	-
1	<i>LEN1 LEN2 (wLength)</i>	2	0100h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>N バイトの ATR</i>	N		カードリセット応答
3+N	<i>CSUM (wChecksum)</i>	1		CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット(エラー)

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	92h	-
1	<i>LEN1 LEN2 (wLength)</i>	2	0200h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>Error Code (bErrorCode)</i>	1	-	エラーコード。付録 A を参考してください。
4	<i>CSUM (wChecksum)</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。

例:

請求 = **62** 01 00 63

応答 = **12** 14 00 3B BE 11 00 00 41 01 38 00 00 00 00 12 34 56 78 01
90 00 73

ATR = 3B BE 11 00 00 41 01 38 00 00 00 00 12 34 56 78 01

6.1.5.2. カードに電源をオフにする (Card Power Off)

このコマンドはリーダにパワーオフのリクエストを送信するために使用されます。

コマンドのフォーマット

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>LEN1 LEN2 (wLength)</i>	2	0100h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>CSUM (wChecksum)</i>	1	62h	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット

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	13h	-
1	<i>LEN1 LEN2 (wLength)</i>	2	0100h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>CSUM (wChecksum)</i>	1	12h	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット (エラー)

オフセット	データフィールド	大きさ	値	数値
0	<i>bMessageType</i>	1	93h	-
1	<i>LEN1 LEN2 (wLength)</i>	2	0200h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>Error Code (bErrorCode)</i>	1	-	エラーコードです。付録 A を参考してください。
4	<i>CSUM (wChecksum)</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。

例:

請求 = 62 01 00 62

応答 = 13 01 00 12

6.1.5.3. カードあるかどうかチェック (Get Card Presence)

このコマンドでカードが挿入されているかどうかを確認します。

コマンドのフォーマット

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>LEN1 LEN2 (wLength)</i>	2	0100h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。 LEN1 は LSB、LEN2 は MSB です。
3	<i>CSUM (wChecksum)</i>	1	64h	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット

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	14h	-
1	<i>LEN1 LEN2 (wLength)</i>	2	0200h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。 LEN1 は LSB、LEN2 は MSB です。
3	<i>STA</i>	1	-	カードステータス: 00 = 未知状態 01 = カードなし 02 = カードある、アクティブされていない 03 = カードある、アクティブされた
4	<i>CSUM (wChecksum)</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット(エラー)

オフセット	データフィールド	大きさ	数値	数値
0	<i>bMessageType</i>	1	94h	-
1	<i>LEN1 LEN2 (wLength)</i>	2	0200h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>Error Code (bErrorCode)</i>	1	-	エラーコードです。付録 A を参考してください。エラーコードです。
4	<i>CSUM (wChecksum)</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。

例:

請求 = 65 01 00 64

応答 = 14 02 00 03 15

6.1.5.4. APDU コマンド(APDU COMMAND)

このコマンドはリーダに APDU コマンドを送信するために使用されます。

コマンドのフォーマット

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>LEN1 LEN2 (wLength)</i>	2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>APDU CMD</i>	N	-	APDU コマンド
3+N	<i>CSUM (wChecksum)</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット

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	11h	-
1	<i>LEN1 LEN2 (wLength)</i>	2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>APDU 応答</i>	N	-	APDU フォーマットデータ
3+N	<i>CSUM (wChecksum)</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット(エラー)

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	91h	-



オフセット	データフィールド	大きさ	数値	説明
1	<i>LEN1 LEN2 (wLength)</i>	2	0200h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>Error Code (bErrorCode)</i>	1	-	エラーコードです。付録 A を参考してください。
4	<i>CSUM (wChecksum)</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。

例:

請求 = 6F 06 00 80 84 00 00 08 65

応答 = 11 0B 00 C1 7A 3B AA D6 5A FA CE 90 00 18

6.1.5.5. APDU2 コマンド(FWv6.01 および以降のバージョン)

このコマンドは、拡張 APDU をサポートする APDU コマンドをリーダーに送信するために使用されます。

コマンドフォーマット

オフセット	データフィールド		大きさ	数値	数値
0	<i>bMessageType</i>		1	67h	-
1	<i>LEN1 LEN2 (wLength)</i>		2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。(最大長さは 263)
3	データ	<i>Param</i>	1	-	<p>パラメーター:</p> <p>短 APDU レベル 00h – デフォルト</p> <p>拡張 APDU レベル 00h – コマンド APDU は、このコマンドで開始および終了します。 01h – コマンド APDU は、このコマンドで開始して、次のコマンド APDU で終了します。 02h – データフィールドは引き続きコマンド APDU を渡し、APDU コマンドを終了します。 03h – データフィールドは引き続きコマンド APDU を渡し、別のデータブロックを付いています。 10h – 空のデータフィールド、次の応答は応答 APDU を渡し続けます</p>
4		<i>APDU CMD</i>	N	-	APDU コマンド (最大長さは 261)
4+N	<i>CSUM (wChecksum)</i>		1	-	CSUM はデータ中のすべての XOR 値を示す。

応答データフォーマット

オフセット	データフィールド		大きさ	数値	数値
0	<i>bMessageType</i>		1	17h	-
1	<i>LEN1 LEN2 (wLength)</i>		2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。 LEN1 は LSB、LEN2 は MSB です。
3	データ	<i>Param</i>	1	-	パラメーター: 短 APDU レベル 00h – デフォルト 拡張 APDU レベル 00h – コマンド APDU は、このコマンドで開始および終了します。 01h – コマンド APDU は、このコマンドで開始して、次のコマンド APDU で終了します。 02h – データフィールドは引き続きコマンド APDU を渡し、APDU コマンドを終了します。 03h – データフィールドは引き続きコマンド APDU を渡し、別のデータブロックを付いています。 10h – 空のデータフィールド、次の応答は応答 APDU を渡し続けます。
4		<i>APDU RSP</i>	N	-	APDU 応答
4+N	<i>CSUM (wChecksum)</i>		1	-	CSUM はデータ中のすべての XOR 値を示す。

応答データフォーマット (WTX)

オフセット	データフィールド		大きさ	値	数値
0	<i>bMessageType</i>		1	18h	-
1	<i>LEN1 LEN2 (wLength)</i>		2	0300h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。 LEN1 は LSB、LEN2 は MSB です。
3	<i>STA</i>		1	-	カードの状態: 00 = 不明 01 = カードが存在しません 02 = カードが存在し、アクティブ化されていません 03 = カードが存在し、アクティブ化されています
4	<i>WTXM</i>		1	-	待機時間遅延乗数

オフセット	データフィールド	大きさ	値	数値
5	CSUM (wChecksum)	1	-	CSUM はコマンド中のすべての XOR 値を示す。

応答データフォーマット(エラー)

オフセット	データフィールド	大きさ	数値	数値
0	bMessageType	1	97h	-
1	LEN1 LEN2 (wLength)	2	0200h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	Error Code (bErrorCode)	1	-	エラーコードです。付録 A を参考してください
4	CSUM (wChecksum)	1	-	CSUM はコマンド中のすべての XOR 値を示す。

例:

600 バイトのデータをカードに送信します

- Command = 67 07 01 01 (261 バイトデータ)チェックサム
Response = 17 02 00 10 チェックサム
- Command = 67 07 01 03 (261 バイトデータ)チェックサム
Response = 17 02 00 10 チェックサム
- Command = 67 50 00 02 (78 バイトデータ)チェックサム
Response = 17 04 00 00 90 00 チェックサム

Receives 600 bytes data from the card

- Command = 67 09 00 00 00 B0 87 00 00 02 58 チェックサム
Response = 17 02 01 01 (256 バイトデータ)チェックサム
- Command = 67 02 00 10 チェックサム
Response = 17 02 01 03 (256 バイトデータ)チェックサム
- Command = 67 02 00 10 checksum
Response = 17 5C 00 02 (88 バイトデータ)チェックサム

6.1.5.6. 直接コマンド(Escape Commands)

このコマンドでリーダーの拡張機能をアクセスできます。

コマンドのフォーマット

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	6Bh	直接コマンドのヘッダ
1	<i>LEN1 LEN2 (wLength)</i>		2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>abData₁</i>	<i>CommandCode</i>	1	-	コマンドヘッダ
4		<i>Len (CommandLength)</i>	1	-	このメッセージ中で、次のフィールドからのバイト数を表して、1 バイトの長さを示します。
5		データ	N	-	0 ≤ N ≤ 255
5+N	<i>CSUM (wChecksum)</i>		1	-	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット

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	15h	直接コマンド応答のヘッダ
1	<i>LEN1 LEN2 (wLength)</i>		2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>abData₂</i>	<i>ResponseCode</i>	1	-	応答のヘッダ
4		<i>Len (CommandLength)</i>	1	-	このメッセージ中で、次のフィールドからのバイト数を表して、1 バイトの長さを示します。
5		データ	N	-	0 ≤ N ≤ 255
5+N	<i>CSUM (wChecksum)</i>		1	-	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット(エラー)

オフセット	データフィールド	大きさ	数値	数値
0	<i>bMessageType</i>	1	95h	-
1	<i>LEN1 LEN2 (wLength)</i>	2	0200h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>Error Code (bErrorCode)</i>	1	-	エラーコードです。付録 A を参考してください。
4	<i>CSUM (wChecksum)</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。

6.1.5.7. カードセットパラメーター (Card Set Parameters)

このコマンドはパワーアップされているカードのパラメーターを変更する時に使われる。

コマンドのフォーマット

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>LEN1 LEN2 (wLength)</i>		2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>abData1</i>	<i>ProtocolNum</i>	1	-	プロトコルのデータ構造: 00h = T=0 プロトコルの構造 01h = T=1 プロトコルの構造
4		<i>ProtocolDataStructure</i>	N	-	プロトコルのデータ構造
4+N	<i>CSUM (wChecksum)</i>		1	-	CSUM はコマンド中でのすべての XOR 値を示す。

応答データフォーマット

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	16h	直接コマンド応答のヘッダ
1	<i>LEN1 LEN2 (wLength)</i>		2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。
3	<i>abData2</i>	<i>ProtocolNum</i>	1	-	プロトコルのデータ構造: 00h = T=0 プロトコルの構造 01h = T=1 プロトコルの構造

オフセット	データフィールド	大きさ	数値	説明
4	<i>ProtocolDataStructure</i>	N	-	プロトコルのデータ構造
4+N	<i>CSUM (wChecksum)</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。

T=0 プロトコルのデータ構造 (ProtocolNum = 0、wLength = 0700h)

オフセット	データフィールド	大きさ	数値	説明
4	<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、ボーレートの変換係数を選択します
5	<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。
6	<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。
7	<i>bWaitingIntegerT0</i>	1	-	T=0 の場合 WI が WWT を定義する時に使われます
8	<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 (ProtocolNum = 1, wLength = 0900h)

オフセット	データフィールド	大きさ	数値	説明
4	<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、ボーレートの変換係数を選択します
5	<i>BmTCKKST1</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。
6	<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。
7	<i>BwaitingIntegerT1</i>	1	-	B7-4 = BWI 値 0-9 有効 B3-0 = CWI 値 0-Fh 有効
8	<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
9	<i>bIFSC</i>	1	-	交渉された IFSC の大きさ
10	<i>bNadValue</i>	1	00h	NAD = 00h だけサポートできます

例: (T0 プロトコル)

リクエスト = 61 07 00 00 11 00 00 0A 00 7D

応答 = 16 07 00 00 11 00 00 0A 00 0A

例: (T1 プロトコル)

リクエスト = 61 09 00 01 96 10 00 45 00 FE 00 54

応答 = 16 09 00 01 96 10 00 45 00 FE 00 23

6.1.6. 相互認証と暗号化プロトコル

ブルートゥースモードで相互認証が成功してから、セッション中の通信プロトコルを暗号化し、転送します。

コマンド	サポートモード	送信側	説明
70h	接続済	ペアリングデバイス	SPH_to_RDR_ReqAuth
71h	接続済	ペアリングデバイス	SPH_to_RDR_AuthRsp

コマンド	サポートモード	送信側	説明
72h	認証済	ペアリングデバイス	SPH_to_RDR_DataReq
20h	接続済	リーダー	RDR_to_SPH_AuthRsp1
21h	接続済	リーダー	RDR_to_SPH_AuthRsp2
22h	認証済	リーダー	RDR_to_SPH_DataRsp

表10 : 相互認証コマンドの概要

6.1.6.1. SPH_to_RDR_ReqAuth

このコマンドは、鍵生成装置に認証を行うために ACR3901T-W1 を要求します。認証が成功してから、ペアリングのキー生成デバイスで顧客マスターキーを変更できます。

認証プロセスの詳細については、[認証セッション](#)を参照してください。

オフセット	データフィールド	大きさ	数値	説明	暗号化
0	<i>bMessageType</i>	1	70h	-	NO
1	<i>LEN1 LEN2 (wLength)</i>	2	0100h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。	
3	<i>wChecksum</i>	1	71h	CSUM はコマンド中でのすべての XOR 値を示す。	

受信したコマンドメッセージはエラーがない場合は、RDR_to_SPH_AuthRsp1 が受信するはずです。



6.1.6.2. RDR_to_SPH_AuthRsp1

このコマンドはペアリングでバイズから送信された SPH_to_RDR_ReqAuth の応答です。

詳しい情報が[認証](#)を参照してください。

オフセット	データフィールド	大きさ	数値	説明	暗号化
0	<i>bMessageType</i>	1	20h	-	NO
1	<i>LEN1 LEN2</i> (<i>wLength</i>)	2	1100h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。	NO
3	<i>abRndNum</i>	16	-	abRndNum[0:15] – 16 バイトの乱数 すべての 16 バイトの乱数は現在に ACR3901T-W1 中でストレージされている顧客マスターキーで暗号化しなければなりません。	YES
19	<i>wChecksum</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。	NO

6.1.6.3. SPH_to_RDR_AuthRsp

このコマンドは認証プロセスの第二段階です。デバイスは SPH_to_RDR_ReqAuth コマンドを ACR3901T-W1 に送信して、エラーがない場合はリーダーは RDR_to_SPH_AuthRsp1 メッセージを戻します。

RDR_to_SPH_AuthRsp1 には顧客マスターキーで暗号化された 16 バイトの乱数が含まれています。ペアリングされたキーの生成デバイスは正しい顧客マスターキーで復号化する必要があります。また、16 バイトの乱数のあとに追加します。それに顧客マスターキーで 32 バイトの乱数全体を復号化して、結果をこのコマンドで ACR3901T-W1 に返すことによって、認証を完了します。

認証プロセスの詳細については、[認証セッション](#)を参照してください。

オフセット	データフィールド	大きさ	数値	説明	暗号化
0	<i>bMessageType</i>	1	71h	-	NO
1	<i>LEN1 LEN2</i> (<i>wLength</i>)	2	2100h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。	NO
3	<i>abAuthData</i>	32	-	abAuthData[0:15] – データ処理サーバに生成された 16 バイトの乱数 abAuthData[16:31] – ACR3901T-W1 に受信された 16 バイトの復号化した乱数 すべての 32 バイトの乱数を AES128 CBC 暗号化モードで顧客マスターキーを使用して、復号化します。	YES
35	<i>wChecksum</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。	NO

受信されたコマンドメッセージはエラーがないで、ACR3901T-W1 に返された乱数も正しい場合、応答は RDR_to_SPH_AuthRsp2 です。



6.1.6.4. RDR_to_SPH_AuthRsp2

このコマンドはペアリングでバイズから送信された *SPH_to_RDR_AuthRsp* の応答です。

詳しい情報が[認証](#)を参照してください。

オフセット	データフィールド	大きさ	数値	説明	暗号化
0	<i>bMessageType</i>	1	21h	-	NO
1	<i>LEN1 LEN2</i> (<i>wLength</i>)	2	1100h	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。	NO
3	<i>abRndNum</i>	16		<i>abRndNum</i> [0:15] – データ処理サーバに受信された 16 バイトの乱数。 すべての 16 バイトの乱数は現在に ACR3901T-W1 中でストレージされている顧客マスターキーで暗号化しなければなりません。	YES
19	<i>wChecksum</i>	1		CSUM はコマンド中でのすべての XOR 値を示す。	NO

6.1.6.5. SPH_to_RDR_DataReq

このコマンドは相互認証プロセスが終了と、ペアリングデバイスから ACR3901T-W1 に送信されます。

ブルートゥースモードで相互認証が成功すると、卡片上電 (Card Power On)からカードセットパラメーター(Card Set Parameters)セッション中の通信プロトコルを暗号化し、転送します。

オフセット	データフィールド	大きさ	数値	説明	暗号化
0	<i>bMessageType</i>	1	72h	-	NO
1	<i>LEN1 LEN2 (wLength)</i>	2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。	NO
3	<i>abEncryptedData</i>	N*16	-	すべての 16 バイトの乱数を AES128 CBC 暗号化モードで顧客マスターキーを使用して、復号化します。	YES
3 + N*16	<i>wChecksum</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。	NO

abEncryptedData の長さは N*16 バイトです。この暗号化のデータ構造は(識別子 + 長さ + ペイロード + チェックサム)です。その中の各バイトは AES128 CBC 暗号化モードで顧客マスターキーを介して、復号化操作を受けます。

AES-128 CBC 暗号化モードでは、初期ベクトルは 16 バイトです(00h)。

元のデータの長さは N*16 より短い場合、暗号化する前に最後に FFh を追加して、長さを 16*N バイトにします。

HID フレーム	長さ(バイト)	説明	
識別子	1	コマンド	本当のデータは <i>abEncryptedData</i> で復号して、ダミーデータを削除します。
長さ	2	長さ {ペイロード+チェックサム}	
ペイロード	0-N	データ	
チェックサム	1	XOR {識別子、長さ、ペイロード}	

例:

相互認証が成功してから、ペアリングデバイスが電気入れるコマンドをリーダーに送信します。コマンドは:

72 11 00 XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

その中:

コマンドヘッダ: 72

電気入れるコマンドの暗号化データ(16 バイト): XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

受信したコマンドメッセージはエラーがない場合は、RDR_to_SPH_DataRsp が受信するはずで。

abData は通信協議の暗号化データです。すべての 16 バイトの乱数を AES-128 CBC 暗号化モードで顧客マスタ



ーキーを使用して、復号化します。

6.1.6.6. RDR_to_SPH_DataRsp

相互認証が成功してから、リーダはこのコマンドをペアリングデバイスに送信します。

ブルートゥースモードで相互認証が成功すると、卡片上電 **Card Power On**)からカードセットパラメーター(Card Set Parameters)セッション中の通信プロトコルを暗号化し、転送します。

オフセット	データフィールド	大きさ	数値	説明	暗号化
0	<i>bMessageType</i>	1	22h	-	NO
1	<i>LEN1 LEN2</i> (<i>wLength</i>)	2	-	このメッセージ中で、次のフィールドからのバイト数を表して、2 バイトの長さを示します。LEN1 は LSB、LEN2 は MSB です。	NO
3	<i>abEncryptedData</i>	N*16	-	すべての 16 バイトの乱数を AES128 CBC 暗号化モードで顧客マスターキーを使用して、復号化します。	YES
3 + N*16	<i>wChecksum</i>	1	-	CSUM はコマンド中でのすべての XOR 値を示す。	NO

6.2. USB 通信プロトコル

ACR3901T-W1 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。

ACR3901T-W1 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、

以下のように概要されています：

1. **制御コマンド** 制御パイプ(デフォルトのパイル)で送信されます。クラス固有の要求や USB 規格の要求が含まれています。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。ACR3901T-W1 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。ACR3901T-W1 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: ACR3901T-W1 にとって、bMaxCCIDBusySlots は 01hに相当です)。

ACR3901T-W1 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	-	ACR3901T-W1 は一つのスロットを持っています。
5	<i>bVoltageSupport</i>	1	-	ACR3901T-W1 は 1.8V、3.0V と 5V のスロットサポート。
6	<i>dwProtocols</i>	4	-	ACR3901T-W1 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	-	最大の ICC 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。
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	-	ACR3901T-W1 は T=1 プロトコルでサポートできる最大の IFSD は 254 です。
32	<i>dwSynchProtocols</i>	4	-	ACR3901T-W1 は同期カードをサポートできません。



オフセット	データフィールド	大きさ	数値	説明
36	<i>dwMechanical</i>	4	-	ACR3901T-W1 は特別な機械的特性をサポートできません。
40	<i>dwFeatures</i>	4	-	ACR3901T-W1 には次の機能があります。 <ul style="list-style-type: none">□パラメーターに従って ICC クロック周波数を自動的に変更する□周波数と FI、DI パラメーターに従ってポーレート自動的に変更するショート APDU および拡張 APDU レベルスイッチング
44	<i>dwMaxCCIDMessageLength</i>	4	-	ACR3901T-W1 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	-	デフォルト 00h
49	<i>bClassEnvelope</i>	1	-	デフォルト 00h
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。

6.2.1. CCID Bulk-OUT メッセージ

6.2.1.1. 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。
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
8	<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。

6.2.1.2. 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_Parameters メッセージです。

6.2.1.3. 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 *RDR_to_PC_SlotStatus* 応答メッセージです。

6.2.1.4. 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	-	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 はこのコマンドで開始および終了します。 0001h – コマンド APDU はこのコマンドで開始し、次の <i>PC_to_RDR_XfrBlock</i> で続きます。 0002h – <i>abData</i> フィールドは引き続きコマンド APDU を渡し、APDU コマンドを終了します。 0003h – <i>abData</i> フィールドは引き続きコマンド APDU を渡し、その後に別のデータブロックが続きます。 0010h – 空の <i>abData</i> フィールド、次の <i>RDR_to_PC_DataBlock</i> は引き続き応答 APDU を渡します
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 (TPDU 交換レベル)

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

6.2.1.5. 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	00000000 0h	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。

6.2.1.6. 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	00000000 h	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。

6.2.1.7. 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>
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、ボーレートの変換係数を選択します</p>
11	<i>bmTCKKST0</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: CCID がこのビットを無視します。</p>
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	-	<p>ICC クロック停止サポート</p> <p>00h = クロックを停止することは許可されていません</p> <p>01h = クロック信号が低い時に停止されます</p> <p>02h = クロック信号が高い時に停止されます</p> <p>03h = クロック信号が低い時または高い時に停止されます</p>

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
-------	----------	-----	----	----

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、ボーレートの変換係数を選択します
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	-	余計な GuardTime (2 文字間の余分な GuardTime は 0–254etu) は FFh である場合、GuardTime を 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。

6.2.1.8. PC_to_RDR_Escape

このコマンドは拡張機能にアクセスするために使用されます。

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	6Bh	-
1	<i>dwLength</i>	4	-	このメッセージの abData 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>abRFU</i>	3	-	保留して将来使います。
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。

6.2.2. CCID Bulk-IN メッセージ

6.2.2.1. RDR_to_PC_DataBlock

このコマンドは ACR3901T-W1 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 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 節で定義されたスロットステータスレジスタ
8	<i>bError</i>	1	-	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>bChainParameter</i>	1	00h	短 APDU レベル、RFU = 00h 拡張 APDU レベル: 00h – 応答 APDU はこのコマンドで開始および終了します。 01h – 応答 APDU はこのコマンドから始まり、続行します。 02h – <i>abData</i> フィールドは引き続き応答 APDU を渡し、応答 APDU を終了します。 03h – <i>abData</i> フィールドは、応答 APDU に続けて別のデータブロックを渡します。 10h – 空の <i>abData</i> フィールド、次の <i>PC_to_RDR_XfrBlock</i> コマンドはコマンド APDU を渡し続けます
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。

6.2.2.2. RDR_to_PC_SlotStatus

このコマンドは ACR3901T-W1 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。
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	-	<u>错误代码</u> 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。

6.2.2.3. RDR_to_PC_Parameters

このコマンドは ACR3901T-W1 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	-	<u>错误代码</u> 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	-	<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</p> <p>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</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>
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。

6.2.2.4. RDR_to_PC_Escape

このメッセージは ACR3901T-W1 によって送信されて、PC_to_RDR_Escape 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	83h	-
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	-	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>bRFU</i>	1	-	将来の使用のために予約 Reserved for Future Use
10	<i>abData</i>	バイト配列	-	CCID から送信されたデータ

7.0. ホストホストプログラミング API

7.1. 周辺機器制御 (Peripherals Control)

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

7.1.1. シリアルナンバーを取得する (Get Serial Number)

このコマンドはリーダーのシリアル番号を取得する時に使われます。

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	02h	Get Serial Number コマンドコード
1		Len (CommandLength)	1	00h	データ中での余分のバイトの数
2		データ	0	-	-

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	82h	Get Serial Number コマンド応答コード
1		Len (CommandLength)	1	-	データ中での余分のバイトの数
2		データ	N	-	シリアルナンバーのバイト数

例:

リクエスト = 02 00

応答 = 82 0A FF FF FF FF FF FF FF FF FF

Serial Number = FF FF FF FF FF FF FF FF FF FF

7.1.2. 乱数を取得する (Get Random Number) コマンド

このコマンドでリーダーの乱数 (AES 暗号化アルゴリズムと認証マスタキーで暗号化された) を取得します。(ブルートゥースモードのみ)

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	03h	Get Random Number コマンドコード
1		Len (CommandLength)	1	00h	データ中での余分のバイトの数
2		データ	0	-	-

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	83h	Get Random Number 応答コード
1		Len (CommandLength)	1	10h	データ中での余分のバイトの数
2		データ	16	-	16 バイトの乱数

例:

リクエスト = 03 00

応答 = 83 10 F2 8F B7 EF BA 43 C4 6B 85 D8 51 7B 84 08 C3 25

7.1.3. 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。

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	04h	Get Firmware Version コマンドコード
1		Len (CommandLength)	1	00h	データ中での余分のバイトの数
2		データ	0	-	-

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	84h	Get Firmware Version 応答コード
1		Len (CommandLength)	1	-	データ中での余分のバイトの数
1		データ	N	-	ファームバージョンのバイト数、フォーマットは“Vx.xx”

例:

リクエスト = 04 00

応答 = 84 08 56 36 2E 30 30 2E 30 30

ファームバージョン (HEX) = 56 36 2E 30 30 2E 30 30

ファームバージョン (ASCII) = “V6.00.00”

7.1.4. マスターキーを書き換える (Rewrite Master Key)

このコマンドはマスターキーを書き換えるに使われます。古いキーを介して、AES 暗号化アルゴリズムで暗号化する必要があります。

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	07h	Rewrite Master Key コマンドコード
1		Len (CommandLength)	1	20h	データ中での余分のバイトの数
2		データ	32	-	元のマスターキーに暗号化された乱数(KeyRstRnd[0:15]) + 16 バイトの元のマスターキーに暗号化された新しいマスターキーをマージする

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	87h	Rewrite Master Key 応答コード
1		Len (CommandLength)	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = 成功 01h = 失敗

例:

詳しい情報は[顧客マスターキーリセットリクエスト](#)を参考してください。

7.1.5. スリープモードオプション(Sleep Mode Option)

このコマンドはデバイスがスリープモードに入る前にの時間間隔を設定する時に使用されます。60 秒後に何も操作しない場合、デフォルトでは、リーダーがスリープモードに入ります。

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	0Dh	Sleep Mode Option コマンドコード
1		Len (CommandLength)	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = 60 秒 (デフォルト) 01h = 90 秒 02h = 120 秒 03h = 180 秒 04h = Disable

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	8Dh	Sleep Mode Option 応答コード
1		Len (CommandLength)	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = 成功 01h = 失敗

例:

リクエストを 90s に設定する = 0D 01 01

応答 = 8D 01 00

7.1.6. デバイスアドレスを取得する (Get Device Address)

このコマンドはデバイスのアドレスを取得するために使用されます。(USB モードのみに使われます。)

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	0Eh	Get Device Address コマンドコード
1		Len (CommandLength)	1	00h	データ中での余分のバイトの数
2		データ	0	-	-

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	8Eh	Get Device Address 応答コード
1		Len (CommandLength)	1	06h	データ中での余分のバイトの数
2		データ	6	-	6 バイトのブルートゥースアドレス

例:

リクエスト = 0E 00

応答 = 8E 06 AA BB CC DD EE FF

デバイスアドレス: AA BB CC DD EE FF

7.1.7. Tx パワーを設定する (Set Tx Power)

ブルートゥースの通信パワーを設定するために使われます。

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	08h	Set Tx Power のコマンドコード
1		Len (CommandLength)	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = -18 dBm (デフォルト) 距離: ~4 米 01h = -12 dBm 距離: ~7 米 02h = -6 dBm 距離: ~16 米 03h = 0 dBm 距離: ~25 米

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	88h	Set Tx Power の応答コード
1		Len (CommandLength)	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = 成功 01h = 失敗

例:

リクエスト = 08 01 00

応答 = 88 01 00

7.1.8. Tx パワーを読み取る (Read Tx Power Value)

ブルートゥースの通信パワーを読み取るために使われます。

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	09h	Read Tx Power のコマンドコード
1		Len (CommandLength)	1	00h	データ中での余分のバイトの数
2		データ	0	-	-

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	89h	Read Tx Power の応答コード
1		Len (CommandLength)	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = -18 dBm (デフォルト) 距離: ~4 米 01h = -12 dBm 距離: ~7 米 02h = -6 dBm 距離: ~16 米 03h = 0 dBm 距離: ~25 米

例:

リクエスト = 09 00

応答 = 89 01 00

7.1.9. カードリセットシミュレーションオプション設定 (Set Card Reset Simulation Option)

このコマンドは、カードリセットイベントを設定するために使用され、ボタンが押されたときに SAM カードを取り外す必要はありません。

コマンドフォーマット

オフセット	データフィールド		大きさ	数値	数値
0	abData1	CommandCode	1	1Ah	Set Card Simulation Option の コマンドコード
1		Len (CommandLength)	1	01h	データの追加バイト数
2		データ	1	-	00h = 禁止 01h = 有効

応答フォーマット

オフセット	データフィールド		大きさ	数値	数値
0	abData2	ResponseCode	1	9Ah	Set Card Simulation Option の コマンドコード
1		Len (CommandLength)	1	01h	データの追加バイト数
2		データ	1	-	00h = 禁止 01h = 有効

例:

請求 = 1A 01 01

応答 = 9A 01 01

7.1.10. カードリセットシミュレーションオプション取得 (Get Card Reset Simulation Option)

このコマンドは、カードリセットシミュレーション機能のステータスを取得するために使用されます。

コマンドフォーマット

オフセット	データフィールド		大きさ	数値	数値
0	abData1	CommandCode	1	1Ah	Get Card Simulation Option のコマンド
1		Len (CommandLength)	1	00h	データの追加バイト数
2		データ	0	-	

応答フォーマット

オフセット	データフィールド		大きさ	数値	数値
0	abData2	ResponseCode	1	9Ah	Get Card Simulation Option のコマンドコード
1		Len (CommandLength)	1	01h	データの追加バイト数
2		データ	1	-	00h = 禁止 01h = 有効

例:

請求 = 1A 00

応答 = 9A 01 01

7.1.11. カードのレスポンスタイムインターバルの設定 (Set Card Response Time Interval)

このコマンドは、カードリセットシミュレーションを有効にして、SAM カードの応答時間間隔を設定します。

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	18h	Set Card Response Time Interval コマンドコード
1		Len (CommandLength)	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = 0s 01h = 500 ms 02h = 1000 ms 03h = 1500 ms (デフォルト) 04h = 2000 ms 05h = 2500 ms 06h = 3000 ms



応答フォーマット

オフセット		データフィールド	大きさ	数値	説明
0	abDat a2	<i>ResponseCode</i>	1	98h	Set Card Response Time Interval 応答コード
1		<i>Len (CommandLength)</i>	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = 成功 01h = 失敗

例:

リクエスト = 18 01 00

応答 = 98 01 00

7.1.12. カードのレスポンスタイムインターバルの読み取り (Get Card Response Time Interval)

このコマンドは、カードリセットシミュレーションを有効にして、SAM カードの応答時間間隔を設定します。

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	19h	Get Card Response Time Interval コマンドコード
1		Len (CommandLength)	1	00h	データ中での余分のバイトの数
2		データ	0	-	

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	99h	Get Card Response Time Interval 応答コード
1		Len (CommandLength)	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = 0s 01h = 500 ms 02h = 1000 ms 03h = 1500 ms (デフォルト) 04h = 2000 ms 05h = 2500 ms 06h = 3000 ms

例:

リクエスト = 19 00

応答 = 99 01 03

7.1.13. ボタンをチェックする (Check button status)

ボタン状態を取得する時にこのコマンドを使用します。

コマンドのフォーマット

オフセット		データフィールド	大きさ	数値	説明
0	abData 1	CommandCode	1	1Bh	Check button status コマンドコード
1		Len (CommandLength)	1	00h	データ中での余分のバイトの数
2		データ	0	-	

応答フォーマット

オフセット		データフィールド	大きさ	数値	説明
0	abData 2	ResponseCode	1	9Bh	Check button status 応答コード
1		Len (CommandLength)	1	01h	データ中での余分のバイトの数
2		データ	1	-	00h = リリース 01h = 押す

例:

リクエスト = 1B 00

応答 (ボタンが押された) = 9B 01 01

7.1.14. 顧客マスターキーをリセットする請求 (Customer Master Key Reset Request)

このコマンドでリーダーは乱数を生成して、顧客マスターキーをリセットします。

コマンドのフォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData1	CommandCode	1	0Fh	Customer Master Key Reset Request コマンドコード
1		Len (CommandLength)	1	00h	データ中での余分のバイトの数
2		データ	0	-	-

応答フォーマット

オフセット	データフィールド		大きさ	数値	説明
0	abData2	ResponseCode	1	8Fh	Rewrite Master Key 応答コード
1		Len (CommandLength)	1	10h	データ中での余分のバイトの数
2		データ	16	-	リーダーが生成した 16 バイトの乱数 (KeyRSTRnd[0:15])

例:

- 乱数を生成する。
顧客マスターキーリセットリクエスト = 0F 00
顧客マスターキーリセットコマンドの応答 = 8F 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
- AES128 CBC モードでは、元のマスターキーを使用して、乱数と新しいマスターキーを暗号化します。上記の操作はアプリケーションの暗号化エンジンに完成されます。暗号化の出力は将来のために、ストレージされます。
乱数: 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
暗号化された乱数: F1 9F D2 D2 BA 1C 22 E1 6D C1 FE 1B 4B 43 D5 30
新しいマスターキー: 11 22 33 44 55 66 77 88 11 22 33 44 55 66 77 88
暗号化された新しいマスターキー: 27 E7 DA BE A6 1E 4B CD 29 F6 9B 36 25 05 8E 41
- マスターキーを書き換える (マスターキーを書き換える (Rewrite Master Key) コマンド を参照してください)。
マスターキーをリセットするコマンドリクエスト = 07 20 F1 9F D2 D2 BA 1C 22 E1 6D C1 FE 1B 4B 43 D5 30 27 E7 DA BE A6 1E 4B CD 29 F6 9B 36 25 05 8E 41
マスターキーをリセットするコマンド = 87 01 00

7.2. メモリカードのコマンドセット

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

7.2.1.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択し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	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)

7.2.1.2. 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)

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)

7.2.1.3. 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)

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)

7.2.1.4. 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)

Pseudo-APDU							
CLA	INS	Byte Address		MEM_L	Byte 1	Byte 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)

7.2.2. メモリカード – 32、64、128、256、512 および 1024 kilobit I2C カード

7.2.2.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択し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	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)

7.2.2.2. 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)

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)

7.2.2.3. 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)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh				

その中:

INS = B0h: 32 kilobit、64 kilobit、128 kilobit、256 kilobit と 512 kilobit の 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)

7.2.2.4. 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)

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

その中:

INS = D0h: 32 kilobit、64 kilobit、128 kilobit、256 kilobit と 512 kilobit 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
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)

7.2.3. メモリカード – ATMEL AT88SC153

7.2.3.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択し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	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)

7.2.3.2. 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)

Pseudo-APDU				
CLA	INS	P1	Byte Address	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)

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)

7.2.3.3. 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)

Pseudo-APDU					
-------------	--	--	--	--	--

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	Byte 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리카ードのメモリアドレス位置
MEM_L	メモリに書き入れていないデータの長さ
MEM_D	メモ리카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)

7.2.3.4. 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)

Pseudo-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리카ードに送信していないシークレットコード
P2	= 0000 00rp ₆ 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: パスワード設定番号、 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。

7.2.3.5. 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)

Pseudo-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)

7.2.3.6. 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)

Pseudo-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)

7.2.4. メモリカード – ATMEL AT88C1608

7.2.4.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択し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	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)

7.2.4.2. 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)

Pseudo-APDU				
CLA	INS	Zone Address	Byte Address	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
= ヒューズを読み取る必要がありません。

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)

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)

7.2.4.3. 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)

Pseudo-APDU								
CLA	INS	Zone Address	Byte Address	MEM_L	Byte 1	Byte 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、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: ヒューズを書く

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)

7.2.4.4. 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)

Pseudo-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。

7.2.4.5. 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)

Pseudo-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)

7.2.4.6. 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)

Pseudo-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)

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

7.2.5.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択し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	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)

7.2.5.2. 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)

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

その中:

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

LSB 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ード中の読み取ら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리카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)

7.2.5.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE4428 と 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)

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 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 2 个字节的虚拟数据

SW1 SW2 = 90 00h(エラーなしの場合)

7.2.5.4. 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)

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

その中:

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

LSB 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 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。

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

例えば、メモリ 0010H から始まる 8 保護ビットを読み取るために、以下の擬似 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								...								
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

7.2.5.5. 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)

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리카ードのメモリアドレス位置である

LSB 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 メモリに書き入れていないデータの長さ

Byte x メモ리카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)

7.2.5.6. 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리카ードのメモリアドレス位置である

LSB 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 メモリに書き入れていないデータの長さ

Byte x バイト値が *Byte Address* から始まるカード内のデータと比較されます。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)

7.2.5.7. PRESENT_CODE_MEMORY_CARD (SLE4428 和 SLE5528)

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する。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しようとする。

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。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。

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

7.2.6.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択し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	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)

7.2.6.2. 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)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FFh	B0h	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리카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)

7.2.6.3. 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)

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** 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)

7.2.6.4. READ_PROTECTION_BITS

このコマンドは始めの 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

7.2.6.5. 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)

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)

7.2.6.6. 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)

7.2.6.7. 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しようとする。

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。

7.2.6.8. 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)

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)

7.2.7. メモリカード – SLE 4406/SLE 4436/SLE 5536/SLE 6636

7.2.7.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択し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	カードタイプ
FFh	A4h	00h	00h	01h	07h

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)

7.2.7.2. 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 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)

7.2.7.3. WRITE_ONE_BYTE_MEMORY_CARD

このコマンドは、挿入されたカードの指定されたアドレスに1バイトを書き込むために使用される。バイトは LSB を初めにカードに書かれている。カードアドレス 0 のビットが 0 バイトの LSB とみなされる。

4 つの異なる書き込みモードは、このカードタイプで使用できる。コマンドデータフィールド内のフラグによって区別され

a. **書き込む**

マンドで指定されたバイトの値が指定されたアドレスに書き込まれて、カードに個人化情報とカウンタ値を書き入れます。

b. **Write with carry**

コマンドで指定されたバイトの値が指定されたアドレスに書き込まれて、コマンドは次の下位カウンタステージを消去するためにカードに送信される。この書き込みモードはカードにカウンタ値を更新するためにのみ使用することができる。

c. **書き入れる時、バックアップ機能を有効にする (SLE 4436、SLE5536 および SLE6636 のみ)**

マンドで指定されたバイトの値が指定されたアドレスに書き込まれて、カードに個人化情報とカウンタ値を書き入れます。バックアップビットを有効にして、カード裂けが発生すると、データの損失を防止することができます。

d. **バックアップ機能の 'Write with carry' コマンドを有効にする (SLE 4436、SLE5536 および SLE6636 のみ)**

コマンドで指定されたバイトの値が指定されたアドレスに書き込まれて、コマンドは次の下位カウンタステージを消去するためにカードに送信される。この書き込みモードはカードにカウンタ値を更新するためにのみ使用することができる。バックアップビットを有効にして、カード裂けが発生すると、データの損失を防止することができます。

以下のモードで、指定されたアドレスのバイトは書き込みの操作を実行する前に消去されないで、メモリビットが '1' を '0' にプログラムしかできません。

SLE4436 カードと SLE5536 カード中で利用可能なバックアップモードは、書き込み動作に有効または無効されます。

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	Byte Address	MEM_L	MODE	BYTE
FFh	D0h	00h		02h		

その中:

Byte Address	メモリカードのメモリアドレス位置
MODE	ライトモードとバックアップオプションを指定する 00h:書き入れる 01h: Write with carry 02h: 書き入れる時、バックアップ機能を有効にする (SLE 4436、SLE5536 および SLE6636 のみ) 03h: バックアップ機能の 'Write with carry' コマンドを有効にする (SLE 4436、SLE5536 および SLE6636 のみ)
BYTE	カードに書き入れていないバイトの値

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)

7.2.7.4. PRESENT_CODE_MEMORY_CARD

メモリカードにシークレットコードを提出して、カードの個人化モードを有効する時に、コマンドが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。

シークレットコードを提出すると、ACR3901T-W1 はプレゼンテーションカウンタを消去しようとしません。アプリケーションソフトウェアによって、独立な 'Write with carry' コマンドを介して行われなければなりません。

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			
					ADDR	Byte 1	Byte 2	Byte 3
FFh	20h	00h	00h	04h	09h			

その中:

ADDR カードプレゼンテーションカウンタのビットアドレス

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

その中:

SW1 SW2 = 90 00h (エラーなしの場合)

7.2.7.5. AUTHENTICATE_MEMORY_CARD (SLE 4436、SLE 5536 および SLE 6636)

SLE 5536 または SLE 6636 からカードの認証証明書を取るために、ACR3901T-W1 は下記の操作をします：

1. コマンドで指定されたカード中の Key 1 または Key 2 を選択する。
2. コマンドで指定された乱数をカードに送信する。
3. カードによって計算した認証データの各ビットに対して指定された CLK パルスの数を生成する。
4. カードから 16 ビットの認証データを読み出す。
5. カードを通常動作モードにリセットする

認証プロセスが二段階で実行される：ステップ1はカードに認証証明書を送信する。ステップ2は、カードによって計算した 2 バイトの認証データを取り戻す。

ステップ1：カードに認証証明書を送信する

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						
					KEY	CLK_CNT	Byte 1	Byte 2	Byte 5	Byte 6
FFh	84h	00h	00h	08h							

その中：

- KEY** 認証証明書を計算するためのキー：
- 00h: Key 1、暗号ブロック連鎖付いていない
- 01h: Key 2、暗号ブロック連鎖付いていない
- 80h: key 1 暗号ブロック連鎖付き (SLE5536 と SLE6636 のみ)
- 81h: key 2 暗号ブロック連鎖付き (SLE5536 と SLE6636 のみ)
- CLK_CNT** CLK のパルス数は、認証証明書の各ビットの計算のためにカードに供給されます。標準値は 160(A0h)です。
- BYTE 1...6** カードの乱数データ

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
61h	02h

その中：

SW1 SW2 = 61 02h (エラーなしの場合)，2 バイトの認証データが準備ができていることを表す。
GET_RESPONSE コマンドによって、認証データを入手する。

ステップ2：認証データを取り戻す(Get_Response)

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

Pseudo-APDU				
FFh	C0h	00h	00h	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)

CERT	SW1	SW2

その中:

CERT カードによって計算された認証データの 16 ビット。BYTE 1 の LSB はカードから読み出された最初の認証ビット。

SW1 SW2 = 90 00h (エラーなしの場合)

7.2.8. メモリカード – SLE 4404

7.2.8.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択し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	カードタイプ
FFh	A4h	00h	00h	01	08h

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)

7.2.8.2. 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 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)

7.2.8.3. WRITE_MEMORY_CARD

このコマンドは、挿入されたカードの指定されたアドレスにデータを書き込むために使用される。バイトは LSB を初めにカードに書かれている。カードアドレス 0 のビットが 0 バイトの LSB とみなされる。

指定されたアドレスのバイトは書き込みの操作を実行する前に消去されないで、メモリビットが '1' を '0' にプログラムしできません。

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 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ないデータの長さ

BYTE カードに書き入れていないバイトの値

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)

7.2.8.4. ERASE_SCRATCH_PAD_MEMORY_CARD

挿入されたカードのスクラッチパッド中のデータを消去する時に、このコマンドを使う。スクラッチパッドメモリ内のすべてのメモリビットが“1”にプログラムされる。

エラーカウンターまたはユーザーゾーンを消去する時、パート **VERIFY_USER_CODE** の説明のように、VERIFY_USER_CODE コマンドを使ってください。

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	Byte Address	MEM_L
FFh	D2h	00h		00h

その中:

Byte Address = スクラッチパッドのメモリバイトアドレス位置
標準値は 02h です

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)

7.2.8.5. VERIFY_USER_CODE

このコマンドはユーザーパスワード(2 バイト)を挿入されているカードに提出します。ユーザーパスワードはカードのストレージのアクセス権限を有効するためです。

以下のアクションが実行される:

1. 指定されたシークレットコードをカードに提出する。
2. 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。
3. プレゼンテーションエラーカウンタを消去する。提出したシークレットコードが成功に認証されて、ユーザーエラーカウンターが消去ことができる。

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	Error Counter LEN	Byte Address	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	04h	08h	02h		

その中:

Error Counter LEN プレゼンテーションエラーカウンタの長さ、ビットの単位です。
Byte Address カード中のキーのバイトアドレス
CODE 2 バイトのユーザーシークレットコード

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

その中:

SW1 SW2 = 90 00h (エラーなしの場合)
= 63 00h (これ以上の再試行がない場合)

注釈: SW1 SW2 = 9000h を受信してから、VERIFY_USER_CODE が正しいかどうかをチェックするために、ユーザーのエラー カウンターを再度読み取るはずでず。ユーザーのエラーカウンタが全部消去されて、'FFh'に等しいされている場合、前回の検証が成功した。

7.2.8.6. VERIFY_MEMORY_CODE

挿入されているカードにメモリコードを提出する時、このコマンドが使用される。| (4 バイト)メモリコードはユーザコードとユーザメモリの再ロードを許可するために使用される。

以下のアクションが実行される:

1. 指定されたシークレットコードをカードに提出する。
2. 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。
3. プレゼンテーションエラーカウンタを消去する。メモリ エラー カウンター中のデータが消去されないことを注意してください。

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	Error Counter LEN	Byte Address	MEM_L	CODE			
					Byte 1	Byte 2	Byte 3	Byte 4
FFh	20h	40h	28h	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

SW1	SW2

その中:

SW1 SW2 = 90 00h (エラーなしの場合)

= 63 00h (これ以上の再試行がない場合)

注釈: SW1 SW2 = 9000 00h を受信してから、VERIFY_MEMORY_CODE が正しいかどうかをチェックするために、アプリケーションゾーンを再度読み取るはずですが、アプリケーションゾーンのデータが全部消去されて、'FFh'に等しいされている場合、前回の検証が成功した。

7.2.9. メモリカード – AT88SC101/AT88SC102/AT88SC1003

7.2.9.1. SELECT_CARD_TYPE

このコマンドはカードリーダーに挿入されて、選択し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	カードタイプ
FFh	A4h	00h	00h	01h	09h

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)

7.2.9.2. 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 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)

7.2.9.3. WRITE_MEMORY_CARD

このコマンドは、挿入されたカードの指定されたアドレスにデータを書き込むために使用される。バイトは LSB を初めにカードに書かれている。カードアドレス 0 のビットが 0 バイトの LSB とみなされる。

指定されたアドレスのバイトは書き込みの操作を実行する前に消去されないで、メモリビットが '1' を '0' にプログラムしできません。

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 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ないデータの長さ

BYTE カードに書き入れていないバイトの値

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)

7.2.9.4. ERASE_NON_APPLICATION_ZONE

このコマンドがアプリケーションゾーンにストアされていないデータを消去する時に使われる。EEPROM メモリが 16 ビットのワードで構造される。独立な 1 ビットのワードを消去しても、ERASE 操作が全てのワードを消去できる。したがって、メモリ中の任意のビットに消去を実行すると、そのメモリのすべての 16 ビットをクリアして、"1"の状態になる。

エラーカウンタまたはアプリケーションゾーンのデータを消去する時、以下のコマンドを参照してください:

1. ERASE_APPLICATION_ZONE_WITH_ERASE コマンドの定義は:

ERASE_APPLICATION_ZONE_WITH_ERASE

2. ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE コマンドの定義は:

ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE

3. *VERIFY_SECURITY_CODE* コマンドの定義は:

VERIFY_SECURITY_CODE

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	D2h	00h		00h

その中:

Byte Address 消去していないワードのメモリバイトアドレスの場所

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)

7.2.9.5. ERASE_APPLICATION_ZONE_WITH_ERASE

このコマンドは以下の状況に適用される:

1. AT88SC101: 擦アプリケーションゾーンのデータを消去して、EC 機能が無効になる。
2. AT88SC102: アプリケーションゾーン 1 のデータを消去する。
3. AT88SC102: アプリケーションゾーン 2 のデータを消去して、EC2 機能が無効になる。
4. AT88SC1003: アプリケーションゾーン 1 のデータを消去する。
5. AT88SC1003: アプリケーションゾーン 2 のデータを消去して、EC2 機能が無効になる。
6. AT88SC1003: アプリケーションゾーン 3 のデータを消去する。

このコマンドで以下の操作を実行する:

1. 指定されたシークレットコードをカードに提出する。
 - a. プレゼンテーションエラーカウンタを消去する。提出したコードが成功に認証されて、アプリケーションゾーンのデータが消去することができる。

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	Error Counter LEN	Byte Address	MEM_L	CODE				
					Byte 1	Byte 2	Byte N
FFh	20h	00h							

その中:

Error Counter LEN プレゼンテーションエラーカウンタの長さ、ビットの単位です。ずっと 00h である

べき。

Byte Address

カード中のアプリケーションゾーンのアドレス。精確な数値が下のチャート
を参照してください。

	Byte Address	LEN
AT88SC101: 擦アプリケーションゾーンを消去して、EC 機能が無効になる。	96h	04h
AT88SC102: アプリケーションゾーン 1 を消去する	56h	06h
AT88SC102: アプリケーションゾーン 2 を消去して、EC2 機能が無効になる。	9Ch	04h
AT88SC1003: アプリケーションゾーン 1 を消去する	36h	06h
AT88SC1003: 擦アプリケーションゾーン 2 を消去して、EC2 機能が無効になる。	5Ch	04h
AT88SC1003: アプリケーションゾーン 3 を消去する	C0h	06h

MEM_L

消去キーの長さ。精確な数値が上のチャートを参照してください。

CODE

N バイトの消去キー

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)

注 釈 : SW1 SW2 = 9000h を受信してから、コマンド ERASE_APPLICATION_ZONE_WITH_ERASE が正しいかどうかをチェックするために、アプリケーションゾーンのデータを再度読み取るはずで、アプリケーションゾーンのデータが全部消去されて、'FFh'に等しい
されている場合、前回の検証が成功した。

7.2.9.6. ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE

このコマンドは以下の状況に適用される:

1. AT88SC101: アプリケーションゾーンのデータを消去して、EC 機能が有効になる。
2. AT88SC102: アプリケーションゾーン 2 のデータを消去して、EC2 機能が有効になる。
3. AT88SC1003: アプリケーションゾーン 2 のデータを消去して、EC2 機能が有効になる。

EC または EC2 機能が有効になってから(すなわち: ECEN または EC2EN ヒューズが損害されていないで、“1”の状態)、以下の操作を実行する:

1. 指定されたシークレットコードをカードに提出する。
2. 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。

3. プレゼンテーションエラーカウンタを消去する。提出したコードが成功に認証されて、アプリケーションゾーンのデータが消去することができる。

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	Error Counter LEN	Byte Address	MEM_L	CODE			
					Byte 1	Byte 2	Byte 3	Byte 4
FFh	20h	80h		04h				

その中:

Error Counter LEN プレゼンテーションエラーカウンタの長さ、ビットで数値は 80h です。

Byte Address カード中のアプリケーションゾーンのアドレス。

	Byte Address
AT88SC101	96h
AT88SC102	9Ch
AT88SC1003	5Ch

CODE 4 バイトの消去キー

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)

 = 63 00h (これ以上の再試行がない場合)

注 釈 : SW1 SW2 = 9000h を受信してから、コマンド ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE が正しいかどうかをチェックするために、アプリケーションゾーンのデータを再度読み取るはずですが、アプリケーションゾーンのデータが全部消去されて、'FFh'に等しいされている場合、前回の検証が成功した。

7.2.9.7. VERIFY_SECURITY_CODE

挿入されているカードにセキュリティコードを提出する時、このコマンドが使用される。| (2 バイト) カードのメモリがアクセスできるように、セキュリティコードはそれが意図されている。

以下のアクションが実行される:

1. 指定されたコードをカードに提出する。
2. 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。
3. プレゼンテーションエラーカウンタを消去する。提出したコードが成功に認証されて、セキュリティコードの試みるカウンタが消去することができる。

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	Error Counter LEN	Byte Address	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	08h	0Ah	02h		

その中:

Error Counter LEN プレゼンテーションエラーカウンタの長さ、ビットの単位です。

Byte Address カード中のキーのバイトアドレス

CODE 2 バイトの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

その中:

SW1、SW2 = 90 00h (エラーなしの場合)

= 63 00h (これ以上の再試行がない場合)

注釈: SW1 SW2 = 9000h を受信した後で、VERIFY_USER_CODE が正しいかどうかを確認するために、セキュリティコードの試みるカウンタ (SCAC) を再度読み取ることができます。SCAC が消去されて、'FFh' に等しいされている場合、前回の検証が成功した。

7.2.9.8. BLOWN_FUSE

このコマンドは挿入されているカードのヒューズを変更する時に使われる。ヒューズは EC_EN のヒューズ、EC2EN のヒューズ、メーカーのヒューズまたは発行者のヒューズ可能です。

注釈: ヒューズを変更することは不可逆過程です。

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	Error Counter LEN	Byte Address	MEM_L	CODE			
					Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of FUS Pin	State of RST Pin
FFh	05h	00h	00h	04h			01h	00h または 01h

その中:

Fuse Bit Addr (2 バイト) ヒューズのビットアドレス。精確な数値が下のチャートを参照してください。

State of FUS Pin FUS pin の状態で、ずっと 01h であるべき。

State of RST Pin RST pin の状態。精確な数値が下のチャートを参照してください。



		Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of RST Pin
AT88SC101	メーカーのヒューズ	05h	80h	01h
	EC2EN のヒューズ	05h	C9h	01h
	発行者のヒューズ	05h	E0h	01h
AT88SC102	メーカーのヒューズ	05h	B0h	01h
	EC2EN のヒューズ	05h	F9h	01h
	発行者のヒューズ	06h	10h	01h
AT88SC1003	メーカーのヒューズ	03h	F8h	00h
	EC2EN のヒューズ	03h	FCh	00h
	発行者のヒューズ	03h	E0h	00h

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. エラーコード

下記のテーブルは ACR3901T-W1 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	説明
01h	チェックサムが無効
02h	データの長さが無効
03h	コマンドフォーマット無効
04h	コマンド無効/未知のコマンド ID
05h	操作エラー
06h	認証必要/認証エラー
07h	バッテリー不足
08h	認証失敗
09h	認証失敗の回数が上限を超えています
0Ah	T1 カード操作エラー

表11 : エラーコード

Android は Google Inc. の商標です。

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

Bluetooth® 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を持っています。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、Windows と Windows Vista 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。