



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1555U



リファレンスマニュアル V2.02

目次

1.0.	紹介	6
1.1.	ACR1555U-A1 バージョン	6
1.2.	記号と略語	6
2.0.	特性	8
3.0.	ACR1555U のアーキテクチャ	9
3.1.	リーダーの機能アーキテクチャ図	9
3.2.	PC/SC ドライバと PICC および SAM 間の通信	10
4.0.	ハードウェアのデザイン	11
4.1.	電池	11
4.1.1.	電池充電と使用マニュアル	11
4.1.2.	バッテリー持続時間	12
4.2.	ブルートゥース	12
4.3.	USB	12
4.3.1.	通信パラメーター	12
4.3.2.	エンドポイント	13
4.4.	非接触スマートカードインターフェース	13
4.4.1.	搬送波周波数	13
4.4.2.	ポーリング	13
4.5.	ユーザーインターフェース	14
4.5.1.	LED	14
4.5.2.	ブザー	15
4.5.3.	ボタン	16
5.0.	ハードウェアデザイン	17
5.1.	リーダーライターモード選択	17
5.2.	ブルートゥース通信	18
5.2.1.	Bluetooth 接続フロー	18
5.2.2.	Profile セレクション	19
5.2.3.	通信配置ファイル	21
5.2.4.	ブルートゥース通信プロトコル	21
5.2.5.	認証	35
5.3.	PCSC API	36
5.3.1.	SCardEstablishContext	36
5.3.2.	SCardListReaders	37



5.3.3.	SCardConnect	38
5.3.4.	SCardControl	39
5.3.5.	SCardTransmit	41
5.3.6.	SCardDisconnect	43
5.3.7.	APDU の流れ	44
5.3.8.	直接なコマンドの流れ	45
5.4.	接触式スマートカードプロトコル	46
5.4.1.	ACOS6-SAM カードコマンド	46
5.5.	非接触スマートカード プロトコル	62
5.5.1.	ATR の生成	62
5.5.2.	APDU、私有 APDU およびカード固有コマンド	65
5.5.3.	PICC の PCSC 私有 APDU（独自の拡張機能付き）	65
5.5.4.	透過コマンド（Pass Through Command）	75
5.5.5.	PCSC 2.0 第 3 部がサポートする APDU コマンド（V2.02 以降）	80
5.5.6.	PICC の専属の私有 APDU	91
5.5.7.	PCSC 準拠のタグをアクセスする（ISO 14443-4）	95
5.5.8.	MIFARE DESFire タグをアクセスする（ISO 14443-3）	96
5.5.9.	サポート PICC ATR	97
6.0.	Escape コマンド	100
6.1.	PICCEscape コマンド	100
6.1.1.	RF 制御（RF Control）[E0 00 00 25 01 ...]	100
6.1.2.	PCD/PICC 状態取得（Get PCD/PICC Status）[E0 00 00 25 00]	101
6.1.3.	ポーリング/ATR オプションの取得（Get Polling/ATR Option）[E0 00 00 23 00]	101
6.1.4.	ポーリング/ATR オプションの設定（Set Polling/ATR Option）[E0 00 00 23 01 ...]	102
6.1.5.	PICC ポーリングタイプ取得（Get PICCPolling Type）[E0 00 01 20 00]	103
6.1.6.	PICC ポーリングタイプ設定（Set PICC Polling Type）[E0 00 01 20 02 ...]	104
6.1.7.	自動 PPS 取得（Get Auto PPS）[E0 00 00 24 00]	105
6.1.8.	自動 PPS 設定（Set Auto PPS）[E0 00 00 24 01 ...]	105
6.1.9.	PICC タイプ取得（Read PICC Type）[E0 00 00 35 00]	107
6.1.10.	PICC- HID キーボードの Escape コマンド	108
6.1.11.	PICC-カードシミュレーションの Escape コマンド	115
6.2.	ICCEscape コマンド	124
6.2.1.	カード電源取得（Get Card Power Configuration）[E0 00 00 0B 00]	124
6.2.2.	カード電源配置設定（Set Card Power Configuration）[E0 00 00 0B 01 ...]	124
6.3.	周辺設備制御と他の Escape コマンド	126
6.3.1.	ファームウェアバージョン取得（Get Firmware Version）[E0 00 00 18 00]	126
6.3.2.	シリアルナンバー取得（Get Serial Number）[E0 00 00 47 00]	126
6.3.3.	USB 記述子内の S/N を設定する（Set S/N in USB Descriptor）[E0 00 00 F0]	127
6.3.4.	ブザー制御の設定-単発（Set Buzzer Control - Single Time）[E0 00 00 28 01 ...]	127
6.3.5.	ブザー制御の設定-重複（Set Buzzer Control - Repeatable）[E0 00 00 28 03 ...]	129

6.3.6.	LED 状態取得 (Get LED Status) [E0 00 00 29 00]	129
6.3.7.	LED 制御設定 (Set LED Control) [E0 00 00 29 01 ...]	130
6.3.8.	UI 操作取得 (Get UI Behaviour) [E0 00 00 21 00]	130
6.3.9.	UI 操作設定 (Set UI Behaviour) [E0 00 00 21 01 ...]	131
6.3.10.	BLE UI 操作取得 (Get BLE UI Behaviour) [E0 00 00 4B 01 05]	131
6.3.11.	BLE UI 操作設定 (Set BLE UI Behaviour) [E0 00 00 4B 02 05 ...]	133
6.3.12.	スリープモードオプションの取得 (Get Sleep Mode Option) [E0 00 00 50 00]	133
6.3.13.	スリープモードオプションの設定 (Set Sleep Mode Option) [E0 00 00 48 ...]	135
6.3.14.	自動シャットダウンオプション取得 [E0 00 00 0F 00]	135
6.3.15.	自動シャットダウンオプション設定 [E0 00 00 0F 01 ...]	136
6.3.16.	即時シャットダウン [E0 00 00 0E 00]	136
6.3.17.	Tx パワー数値取得 (Get Tx Power Value) [E0 00 00 51 00]	136
6.3.18.	Tx パワー数値設定 (Set Tx Power Value) [E0 00 00 49 ...]	137
6.3.19.	MAC アドレス取得 (Get MAC Address) [E0 00 00 43 00]	137
6.3.20.	BLE 放送名称取得する (Get BLE Advertising Name) [E0 00 00 44 00]	138
6.3.21.	バッテリー取得 (Get Battery Level) [E0 00 00 52 00]	139
6.3.22.	BLE バインドレコードの削除 (Remove BLE Bonding Record) [E0 00 00 5B 00]	139
6.3.23.	BLE 通信モード読み取る (Read BLE Communication Mode)	140
6.3.24.	BLE 通信モード設定 (Set BLE Communication Mode)	140
6.3.25.	顧客マスターキーリライター (Customer Master Key Rewrite)	142
付録 A. NDEF メッセージ		143
付録 B. ロットビット状態とスロットビットエラー		144

図示一覧表

図 1 :	ACR1555U のアーキテクチャ	10
図 2 :	BLE スプロトコル・スタック	10
図 3 :	リーダーライターモード	17
図 4 :	Bluetooth 接続フロー	18
図 5 :	認証手順	35
図 6 :	ACR1555U の APDU 流れ	44
図 7 :	ACR1555U 直接コマンドの流れ	45
図 8 :	透過セッションのフロー図	80

チャート一覧表

表 1:	変更履歴	6
表 2:	記号と略語	7



表 3 : 推定のバッテリー持続時間	12
表 4 : USB インターフェース配線	12
表 5 : Bluetooth モードでの LED 表示	14
表 6 : USB モードでの LED 表示	15
表 7 : ACR1555UBluetooth サービス	19
表 8 : ACR1555U のサービスハンドルと UUID メッセージリスト	20
表 9 : コマンドコードの概要	21
表 10 : 応答コードの概要	22
表 11 : カード通知コード概要	22
表 12 : MIFARE Classic 1K カードのメモリマップ	70
表 13 : MIFARE Classic 4K カードのメモリマップ	70
表 14 : MIFARE Ultralight カードのメモリマップ	71
表 15 : NFC フォーラムタイプ 2 ラベルのメモリマップ (2000 バイト)	116
表 16 : FeliCa カードのメモリマップ (160 バイト)	117
表 17 : スロットビットステータスレジスタ	144
表 18 : スロットビットエラーレジスタ (bmCommandStatus = 1)	145

1.0. 紹介

ACR1555U NFC Bluetooth®リーダライタは ISO 14443 第 1-4 部と ISO 18092 規格に適合し、非接触カード、MIFARE®カード、FeliCa™カード、NFC タブ及び ISO 7816 A、B と C 類（5 V、3 V と 1.8 V）SAM カードをサポートしています。

このリーダライタは Bluetooth®5.2 標準的なシングルモード操作をサポートする iOS、Android™、Linux®、Windows®などの各種オペレーティングシステムに対応できる USB Type-C デバイスである。

ACR1555U は 450 mAh 3.7 V リチウムイオン電池を用いて電力を供給します。設備の運転状態を表示するために、3 つの LED ランプと 1 つのブザーを装備しています。また 2 つのボタンがそれぞれに電池、設備状態及び Bluetooth®状態を制御します。

1.1. ACR1555U-A1 バージョン

バージョン ⁽¹⁾	公開日	変更内容
REV 1	2024 年 11 月	- 初期バージョンリリース - シリアル番号：RR629-XXXXXX - リファレンスマニュアル V1.XX またはそれ以前のバージョンに対応
REV 2	2026 年 1 月	- シリアル番号：RR662-XXXXXX - 動作モードでのバッテリー寿命が延長 - リファレンスマニュアルが V2.00 にアップグレード

表 1: 変更履歴

1.2. 記号と略語

略語	説明
ATR	属性要求と属性応答（Attribute Request and Attribute Response）
DEP	データ交換プロトコル要求及びデータ交換プロトコル応答（Data Exchange Protocol Request and Data Exchange Protocol Response）
DSL	要求の選択解除と応答の選択解除（Deselect Request and Deselect Response）
PSL	パラメータ選択要求とパラメータ選択応答（Parameter Selection Request and Parameter Selection Response）
RLS	要求リリースと応答リリース（Release Request and Release Response）
WUP	要求ウェイクアップと応答ウェイクアップ（Wakeup Request and Wakeup Response）



略語	説明
DID	設備 ID (Device ID)
BS	ビット周期送信 (Sending bit duration)
BR	ビット周期受信 (Receiving bit duration)
PP	プロトコルパラメーター (Protocol Parameters)

表 2: 記号と略語

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
- CCID準拠
- スマートカードリーダー：
 - 非接触インターフェース：
 - 読み書き速度が 53 kbps (ISO 15693 カード) および 848 kbps (ISO 14443 カード)
 - 内蔵アンテナを使って、ACR122U の通信距離は最大 5cm です。(タグのタイプに応じて)
 - ISO 15693 カードサポート
 - ISO 14443 A および B カード、MIFARE®シリーズ、FeliCa、5 タイプのすべての NFC タグ (ISO/IEC 18092) もサポートしています。
 - 衝突防止機能内蔵
 - 拡張の APDU サポート (最大 64 KB)
 - SAM インターフェース：
 - 1つ SAMカードスロット
 - ISO7816クラスA のSAMカードサポート
- 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
 - CT-API サポート (PC/SC 上のレイヤーによるパッケージ)
- 内蔵されている周辺機器：
 - 3×ユーザー制御可能な LED (青色、黄色、赤&緑の二色 LED)
 - ユーザー制御可能なブザー
 - バッテリ、デバイス、Bluetooth®ステータスを制御するボタン
- USBファームウェアアップグレードサポート¹
- Android™ 4.3と以降のバージョンサポート²
- iOS とiPadOS 12とその以降のバージョンサポート³
- 以下の規格に準拠：
 - IEC/EN 62368
 - CE
 - UKCA

¹PC リンクモードに適用

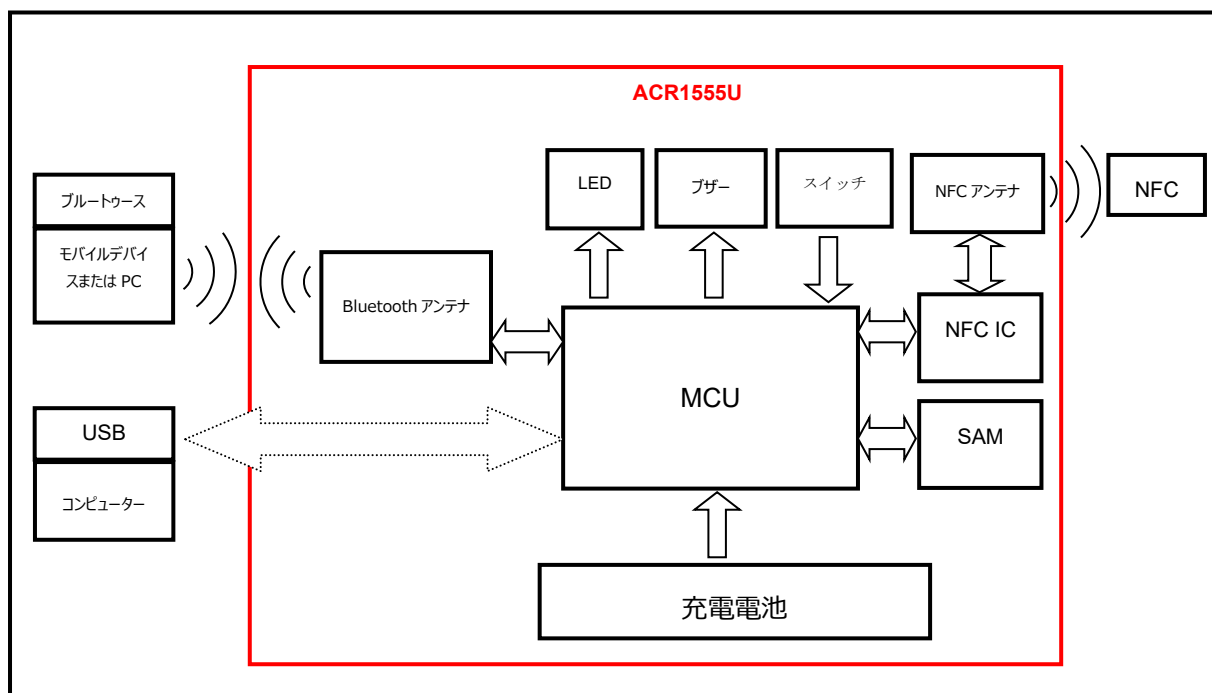
²ACS 定義された Android ライブラリ使用

³ACS 定義される iOS または iPadOS ライブラリを使用する

- FCC
- VCCI
- RoHS
- REACH
- Bluetooth® BQB
- TELEC (日本)
- Microsoft® WHQL
- KCC (韓国)
- ISO 14443
- ISO 15693
- ISO 7816
- PC/SC
- CCID
- WEEE

3.0.ACR1555U のアーキテクチャ

3.1. リーダーの機能アーキテクチャ図



3.2. PC/SC ドライバと PICC および SAM 間の通信

ACR1555U が CCID プロトコルを使用して PC とデータ通信します。PICC と SAM 間の通信は PC/SC 規格に準拠しています。

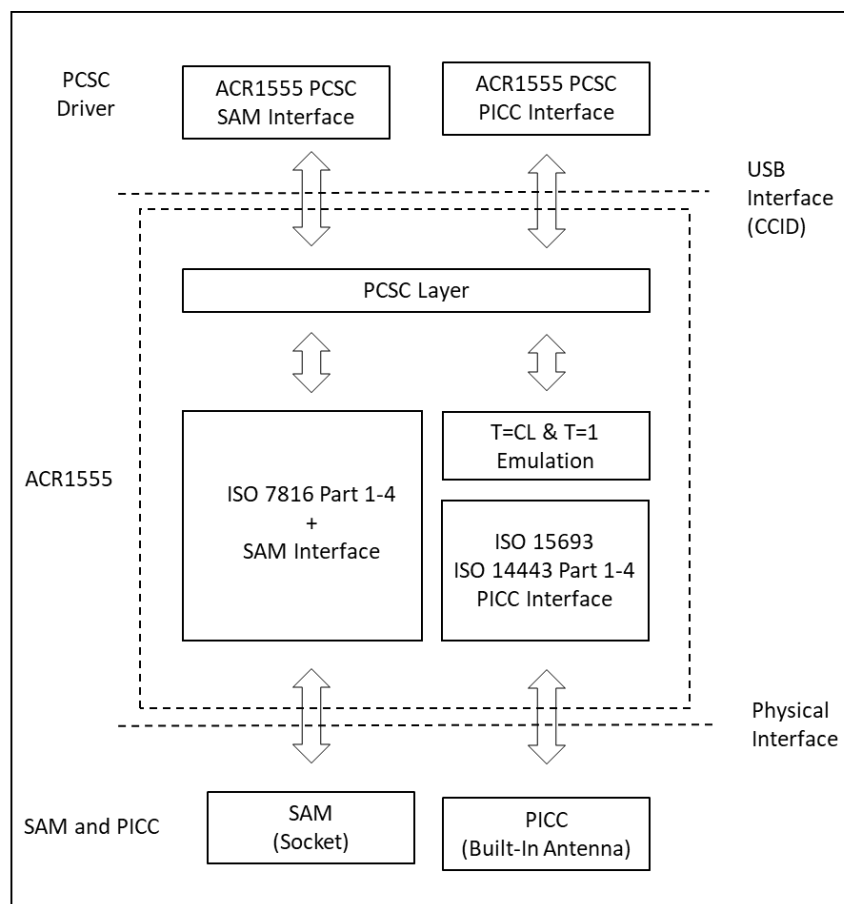


図1 : ACR1555U のアーキテクチャ

ブルートゥース低エネルギープロトコル・スタック・アーキテクチャは次とおりです :

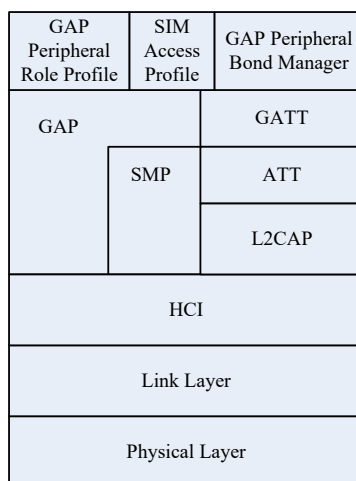


図2 : BLE スプロトコル・スタック

4.0. ハードウェアのデザイン

4.1. 電池

ACR1555U は 450 mA の充電式リチウムイオン電池を使用しています。

4.1.1. 電池充電と使用マニュアル

ACR1555U バッテリーの電力が切れた場合は、電源オフ、USB、または Bluetooth モードで電源コンセントに接続して充電できます。

ACR1555U のバッテリーが切れた場合、デバイスの電源をオフにした状態で充電することも、USB または Bluetooth 操作中に電源に接続して充電することもできます。

充電

- **公式 USB ケーブルを使用する：**

ACR1555 の充電には、必ず公式の USB ケーブルを使用してください。互換性のない充電アクセサリを使用すると、故障や損傷の原因になることがあります。

- **おすすめの充電器：**

安全性と最適な性能を確保するために、出力が 5 V、500 mA の片口充電器を使用することをお勧めします。

- **初回充電：**

初めて設備を使用する場合は、フル充電してください。

- **定期充電：**

使用しない間でもバッテリーがゆっくりと放電するため、毎回使用する前に充電して、ACR1555 が常に使用可能であることを確保してください。

- **充電後：**

充電が完了したら、コンセントから充電器を抜くか、ACR 1555 から USB ケーブルを抜いてください。これはバッテリー寿命の延長に役立ちます。

安全注意事項

- **防水防湿：**

ACR1555 を水や湿気から離してください。この設備は防水性能を備えておらず、液体に接触すると故障や永久損傷を招く可能性があります。

- **極端な温度を避ける：**

車内や直射日光などの高温環境に長時間放置しないでください。

電池の寿命

- **使用影響：**

バッテリー寿命は貯蔵方式、使用条件、環境要因に依存します。

ワンポイント： バッテリー寿命を最大限に延ばすために、過充電や完全放電を避けてください。電源が切れていなくても、定期的に充電し、極端な温度環境にデバイスを置かないようにしてください。

4.1.2. バッテリ持続時間

バッテリーの持続時間はデバイスの使用状況によって異なります。様々な作業条件に応じて、バッテリー持続時間の推定値は以下のようになります。

モード	推定のバッテリー持続時間
動作モード	18 時間 * ⁽¹⁾
自動電源オフモード	18 日 * ⁽²⁾

表3：推定のバッテリー持続時間

注意： 実際の結果は、使用するスマートカードの種類や動作環境によって異なる場合があります。上記の計算は、ACR1555U-A1 REV2 RR662-XXXXXX ⁽⁴⁾ でのテスト結果に基づいています。

1 動作モード：Bluetooth モードでは、デバイスが 1 日 10 回（各 1 分間）の操作を行い、1 日合計 10 時間 Bluetooth 接続を維持します。

2 自動電源オフモード：Bluetooth モードでは、デバイスは 1 日 10 回（各 1 分）の操作を行います。各操作後、デバイスは 5 分以内に自動的に電源がオフになります。

3 ハードウェアバージョンはリーダーの裏蓋を開けることで確認できます。詳細は 1.1 章、表 1「変更履歴」をご参照ください。

4.2. ブルートゥース

ACR1555U は、Bluetooth でデバイスをコンピュータおよびモバイルデバイスとペアリングします。

4.3. USB

ACR1555U が USB 規格に準拠した USB を介して PC と接続します。

4.3.1. 通信パラメーター

ACR1555U は USB 2.0 規格 2.0 に従って USB インタフェースを通じてコンピュータと接続を確立します。USB 全速モードをサポートし、レートが 12 Mbps になります。

ピン	信号	機能
1	V _{BUS}	カードに +5 V の電源を供給
2	D-	ACR1555U が PC との間で差動信号でデータを伝送する
3	D+	ACR1555U が PC との間で差動信号でデータを伝送する
4	GND	参考用の電圧レベル

表4：USB インターフェース配線



注：ACR1555U が USB インタフェースを介して正しく機能するためには、**ACS 独自のデバイスドライ**または **Microsoft CCID CCID ドライ**のいずれかをインストールする必要があります。詳細は『デバイスドライバインストールガイド』を参照してください。

4.3.2. エンドポイント

ACR1555U 下記のエンドポイントを介して、ホスト PC と通信します：

Control Endpoint 設定と制御用

PICC：

EP1 Bulk OUT ホスト PC から ACR1555U PICC インターフェースにコマンドを送信する（パケットサイズ 64 バイト）

EP1 Bulk IN ACR1555U PICC からホスト PC に応答を送信する（パケットサイズ 64 バイト）

EP2 Interrupt IN ACR1555U PICC からホスト PC にカード状態のメッセージを送信する（パケットサイズ 8 バイト）

SAM：

EP3 Bulk OUT ホスト PC から ACR1555U SAM インターフェースにコマンドを送信する（パケットサイズ 64 バイト）

EP3 Bulk IN ACR1555U SAM からホスト PC に応答を送信する（パケットサイズ 64 バイト）

4.4. 非接触スマートカードインターフェース

ACR1555U と非接触カードとの間のインタフェースが ISO 14443 規格に準拠しており、ACR1555U の実用的な機能を強化するためにいくつかの制限または向上が行われています。

4.4.1. 搬送波周波数

ACR1555U の搬送波周波数は 13.56MHz です。

4.4.2. ポーリング

ACR1555U に入る非接触カードを自動的に検出します。この機能は ISO 14443-4 の A と B タイプのカード、Topaz カード、MIFARE カード、ISO15693 カード、FeliCa と NFC タグをサポートします。

4.5. ユーザーインターフェース

4.5.1. LED

LED インジケータは、Bluetooth モード、電源、USB モードの状態を表示するために使用されます。青色 LED が Bluetooth モードの状態を表示するために使用され、緑色 LED がデバイスの電源状態を表示するために使用され、赤色 LED がバッテリー状態を表示するために使用され、黄色 LED が PICC 状態を表示するために使用されます。

モード	色	LED ステータス	状態
ブルートゥースモード	青色 //(LED1)	OFF	リーダーライター電源オフ Bluetooth デバイスとペアリングしていない USB モード
		速く点滅する	ユーザーのペアリング確認を待つ（モードボタンを 1 回押す必要がある）
		点滅	信号放送 デバイスのペアリング待ち
		ゆっくり点滅 (0.2 Hz)	スタンバイモード
		点灯	Bluetooth デバイス接続完了
	赤 //(LED2)	OFF	バッテリー充電終了 リーダーライターは USB のみで電源支給/ 電池の電圧が 3.5 V より高くて、USB 電源ありません
		ゆっくり点滅 (0.2Hz)	バッテリー残量不足（30%未満）
		点灯	充電中
	オレンジ ⁴ (LED2)	点灯	デバイスに電源が入り、充電中
	緑 //(LED2)	OFF	設備電源オフ / スタンバイモード
		点灯	設備電源入り
	黄色 //(LED3)	OFF	RF オフ
		速く点滅する	スマートカードがリーダーと読み/書きます
		点灯	カードが存在して、リーダーがコマンドを待っています：

表5：Bluetooth モードでの LED 表示

⁴緑色と赤色が同時に点灯する

モード	色	LED ステータス	状態
USB モード	青色 //(LED1)	OFF	USB モード
	赤 //(LED2)	OFF	バッテリー充電終了 リーダ/ライタは USB のみで電源支給/ 電池の電圧が 3.5V より高く、USB 電源がありません
		ゆっくり点滅	バッテリー残量不足 (30%未満)
		点灯	充電中
	オレンジ ⁵ (LED2)	点灯	デバイスに電源が入り、充電中
	緑 (LED2)	OFF	設備電源オフ
		点灯	設備電源入り
	黄色 //(LED3)	OFF	カードが存在しなくて、RF オフ
	黄色 //(LED3)	速く点滅する	スマートカードがリーダと読み/書きます
		点灯	カードが存在して、リーダがコマンドを待っています :
		OFF	カードが存在しなくて、RF オフ

表6 : USB モードでの LED 表示

4.5.2. ブザー

カードポーリング、Bluetooth 接続、スリープモード、低バッテリー状態をユーザに通知するために使われます。

ブザー操作	イベント
1 回ビーブ	1.リーダが電源と接続済 2.カードが検出された/カードが外した 3.USB モードから Bluetooth モードへの切り替え (長いビーブ音)
2 回ビーブ	リーダが電源と接続済、バッテリー残量不足
3 回ビーブ	電源接続切れた

⁵緑色と赤色が同時に点灯する

4.5.3. ボタン

ACR1555U には BLE ボタンとスイッチボタンの 2 つのボタンがあります。

ボタン	条件	モード	ボタン状態	説明
モードボタン	-	BLE モード（ペアリング）	短く押す	Bluetooth バインドの確認
	PC USB 挿入	USB モード	長く押す	BLE モードに切り替える
	ブルートゥース 接続済	BLE モード	長く押す	USB モードに切り替える ⁽¹⁾
スイッチ	リーダーオフ	任意	長く押す	オン
	リーダーオン		長く押す	オフ
			短く押す	リカバリオペレーション

⁽¹⁾ Requires firmware version 1.03.02 or higher.

5.0. ハードウェアデザイン

5.1. リーダーライターモード選択

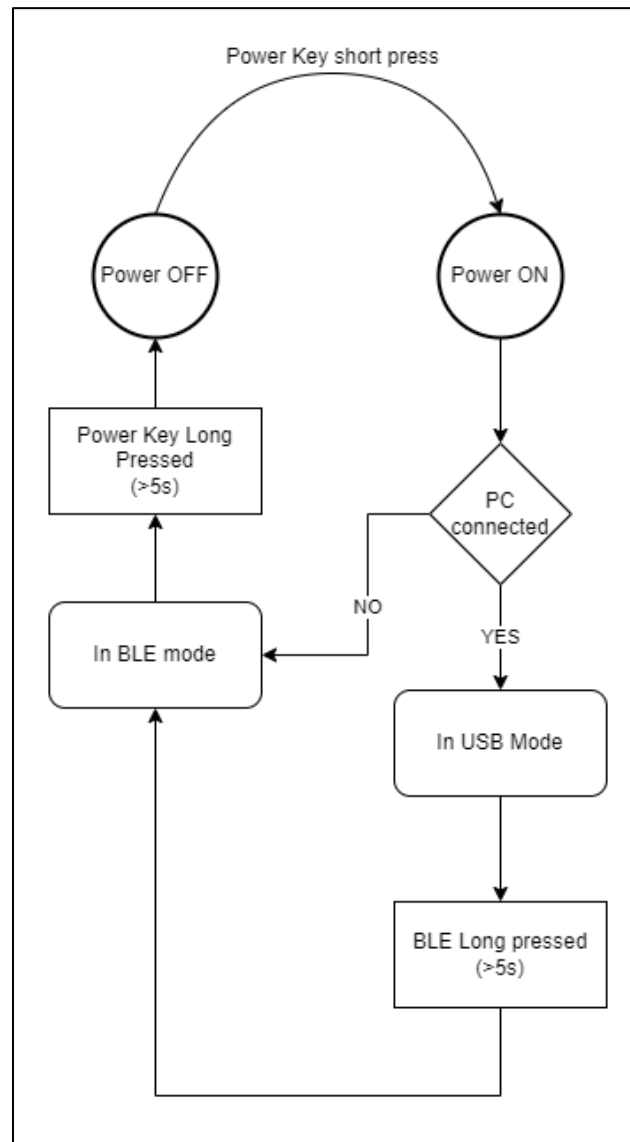


図3：リーダーライターモード

リーダライタが起動すると、まず USB ポートとの接続状況を検査します。ポートに 5 V の電圧が検出されると、リーダライタは USB モードを起動します。正常に起動すると USB モードのままになります。USB モードが起動できなかった場合、リーダライタは Bluetooth ブロードキャストモードをオンにします。Bluetooth モードから USB モードに切り替える必要がある場合は、ユーザーはデバイスの電源を切り、再び電源を入れて、USB デバイスを再確認する必要があります。

5.2. ブルートゥース通信

5.2.1. Bluetooth 接続フロー

下記は Bluetooth の接続フローです：

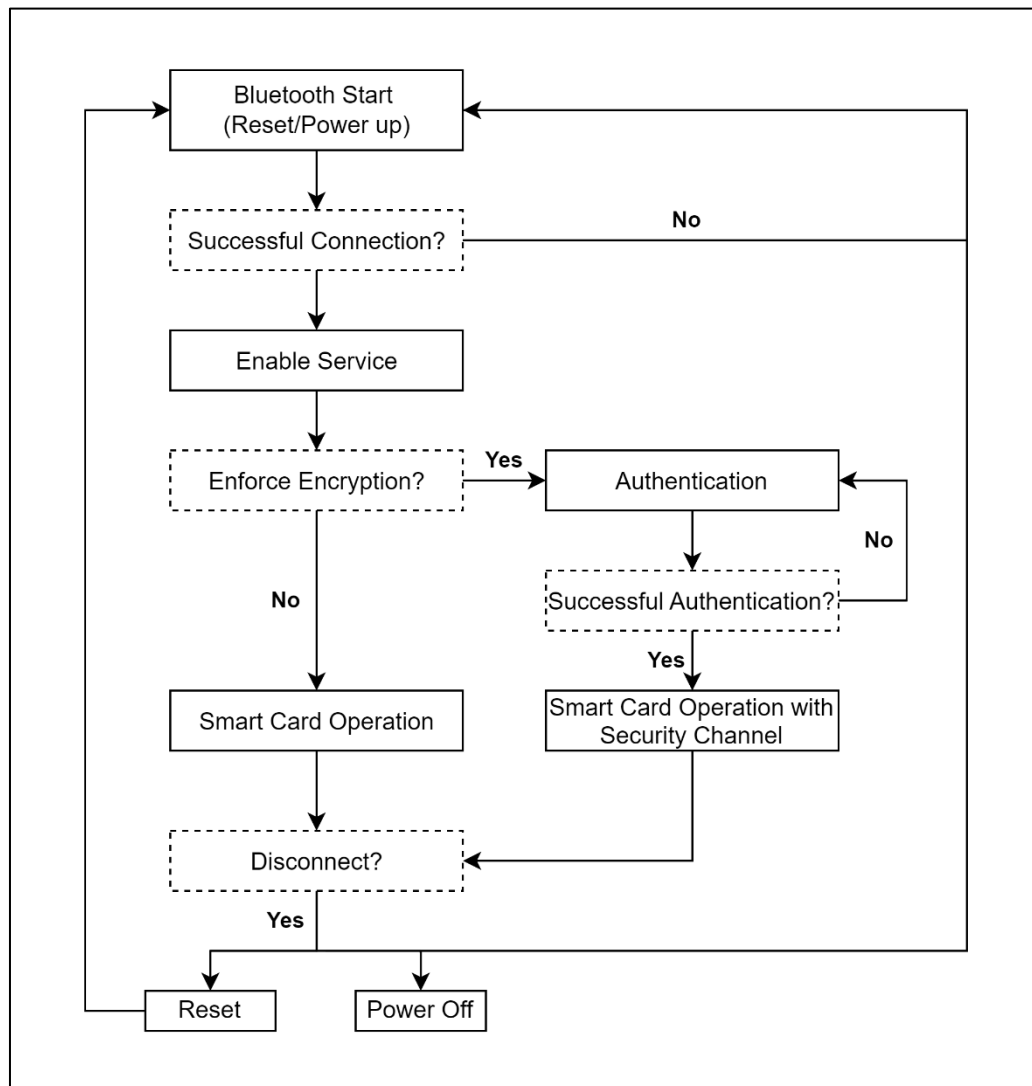


図4：Bluetooth 接続フロー

5.2.2. Profile セレクション

ACR1555U はインターフェースを送信するためのデータとしての 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 つのチャンネルを介してコマンド通信のカスタムサービスを採用しています。1 つ目のチャンネルはコマンド要求に使用され、2 つ目のチャンネルはコマンド応答に使用され、3 つ目のチャンネルはカード通知に使用されます。

また、リーダーライターが Bluetooth モードの場合、現在の消費電力が大幅に増加するため、標準バッテリーサービスを使用して、ペアリングデバイスの現在のバッテリー状態を通知します。電池状態が変わった場合、リーダーは、特定のチャンネルを介してペアリングデバイスに通知します。操作を簡単にするために、バッテリーが 3 つのグループに分けられています：残量十分 (≥ 3.4 V)、残量不足 (< 3.4 V と ≥ 3.1 V) およびバッテリーなし (< 3.1 V)。

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。この情報を手動で読み取り、またはアプリケーションの要求によって読み取ることができます。情報の中にはモデル番号、シリアル番号、ファームウェアバージョン、およびデバイス名が含まれます。

サービス	UUID	パイプ
スマートカード	00003971-817C-48DF-8DB2-476A8134EDE0	コマンド請求
	00003972-817C-48DF-8DB2-476A8134EDE0	コマンド応答
	00003973-817C-48DF-8DB2-476A8134EDE0	カード通知
電池	2A19	電池残量
デバイス情報	2A23	システム ID
	2A24	型番
	2A25	シリアルナンバー
	2A26	ファームウェアのバージョン
	2A27	ハードウェアのバージョン
	2A29	メーカー名称

表7：ACR1555U Bluetooth サービス

属性名称	UUID	ハンドル
DeviceName	2A00	06h
送信 (リーダーライター → ペアリングデバイス)	00003971-817C-48DF-8DB2-476A8134EDE0	28h
受信 (ペアリングデバイス → リーダーライター)	00003972-817C-48DF-8DB2-476A8134EDE0	2Bh
カード通知する (リーダー → ペアリングデバイス)	00003973-817C-48DF-8DB2-476A8134EDE0	2Fh



属性名称	UUID	ハンドル
ABatteryLevel	2A19	20h
Manufacturer	2A29	19h
SerialNumber	2A25	11h
FW_Version	2A26	13h
ModelNumber	2A24	0Fh

表8 : ACR1555U のサービスハンドルと UUID メッセージリスト

5.2.3. 通信配置ファイル

通信配置ファイルが次のようになります。

開始バイト+スロットビット+長さ+保留（1バイト）+データブロック+チェックサム+終了バイト

データフィールド	大きさ（バイト）	説明
開始バイト	1	値：55h
スロットビット	1	カードスロット 00h：PICC、カードスロット 01h SAM
長さ	2	長さは Datablock データフィールド中のバイト数を示す
保留	1	保留
ホストのシリアルナンバー	1	ホストが新しいフレームを送信するときに 1 バイト増やす
リーダーライターシリアルナンバー	1	リーダーライターが新しいフレームを送信するときに 1 バイト増やす
データブロック	N	データ（CCID に一致するメッセージ体）
チェックサム	1	スロットビット、長さ、フレームタイプ、ホスト&リーダーライターシリアル番号、およびデータメインの排他的 OR（XOR）値
終了バイト	1	値：AAh

5.2.4. ブルートゥース通信プロトコル

ACR1555U は、予め定義されたプロトコルを採用して、ブルートゥース・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。このプロトコル、CCID コマンドチャンネルと応答チャンネル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	明文 認証済	ペアリングデバイス	PICC 電気入れる
63h	明文 認証済	ペアリングデバイス	PICC パワーオフ
65h	明文 認証済	ペアリングデバイス	カードのステータスを取得
6Fh	明文 認証済	ペアリングデバイス	APDU 交換
6Bh	明文 認証済	ペアリングデバイス	Escape コマンド
61h	明文 認証済	ペアリングデバイス	パラメータを設定する

表9：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
80h	明文 認証済	リーダー	データブロックに対する応答
81h	明文 認証済	リーダー	カードスロットステータスに対する応答
82h	明文 認証済	リーダー	パラメーターに対する応答
83h	明文 認証済	リーダー	Escape コマンドに対する応答
53h	明文 認証済	リーダー	エラーに対する応答

表10：応答コードの概要

コマンド	サポートモード	送信側	説明
50h	明文 認証済	リーダー	カードステータスを通知する
52h	明文 認証済	リーダー	スリープモードに入るのを通知する

表11：カード通知コード概要

5.2.4.1. PC_to_RDR_IccPowerOn

スロットを活性化して、カードから ATR を返す。

オフセット	データフィールド	大きさ	数値	説明
0	bMessageType	1	62h	-
1	dwLength	4	00 00 00 00h	メッセージの余分なバイトのサイズ。
2	bSlot	1	00-01h	このコマンドのスロット番号を識別します。 00h : PICC 01h : SAM
5	bSeq	1	00-FFh	コマンドのシーケンス番号
6	bPowerSelect	1	00h-02h	ICC 用の電圧 00H - 自動電圧の選択 01H - 5 ボルト 02H - 3 ボルト
7	abRFU	2	0000h	保留して将来使います。

このメッセージの応答は RDR_to_PC_DataBlock メッセージです。返されたデータはリセット応答（ATR）です。

5.2.4.2. PC_to_RDR_IccPowerOff

アクティベーション 状態をキャンセルします

オフセット	データフィールド	大きさ	数値	説明
0	bMessageType	1	63h	-
1	dwLength	4	00000000h	メッセージの余分なバイトのサイズ。
5	bSlot	1	00-01h	このコマンドのスロット番号を識別します。 00h : PICC 01h : SAM
6	bSeq	1	00-FFh	コマンドのシーケンス番号
7	abRFU	3	000000h	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。

5.2.4.3. PC_to_RDR_GetSlotStatus

現在のカードスロットの状態を取得します。

オフセット	データフィールド	大きさ	数値	説明
0	bMessageType	1	65h	-
1	dwLength	4	00 00 00 00h	メッセージの余分なバイトのサイズ。
5	bSlot	1	00-01h	このコマンドのスロット番号を識別します。 00h : PICC 01h : SAM
6	bSeq	1	00-FFh	コマンドのシーケンス番号
7	abRFU	3	000000h	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。

5.2.4.4. PC_to_RDR_XfrBlock

ICC にデータブロックを転送します。



オフセット	データフィールド	大きさ	数値	説明
0	bMessageType	1	6Fh	-
1	dwLength	4	00 00 00 00 ~ 00 00 01 E7h	このメッセージの abData データフィールドの サイズ データドメインはリトルエンドフォーマットで格 納されます
5	bSlot	1	00-01h	このコマンドのスロット番号を識別します。 00h : PICC 01h : SAM
6	bSeq	1	00-FFh	コマンドのシーケンス番号
7	bBWI	1	00-FFh	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。

オフセット	データフィールド	大きさ	数値	説明
8	wLevelParameter	2	-	フィールドはリトルエンドフォーマットで格納されます TPDU レベル、RFU、= 0000h 短 APDU レベル、RFU、= 0000h 拡張 APDU レベル： APDU がコマンドで開始または終了するかどうかを標識します。 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 データフィールドは継続にコマンド APDU を転送し、APDU コマンドを終了します。 0003h abData データフィールドが継続にコマンド APDU を転送し、後にもう 1 つのデータブロックを追跡し、 0010h 空の abData データフィールド、次の RDR_to_PC_DataBlock が応答 APDU を転送し続ける
10	abData	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。

5.2.4.5. PC_to_RDR_Escape

拡張機能をアクセスする

オフセット	データフィールド	大きさ	数値	説明
0	bMessageType	1	6Bh	-
1	dwLength	4	00 00 00 00 ~ 00 00 00 FFh	このメッセージの abData データフィールドのサイズ。 データドメインはリトルエンドフォーマットで格納されます
5	bSlot	1	00h - 01h	このコマンドのスロット番号を識別します。 00h : PICC 01h : SAM
6	bSeq	1	00-FFh	コマンドのシーケンス番号
7	abRFU	3	000000h	保留して将来使います。
10	abData	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_Escape メッセージです。

5.2.4.6. PC_to_RDR_SetParameters

カードスロットのパラメータを設定します。

オフセット	データフィールド	大きさ	数値	説明
0	bMessageType	1	61h	-
1	dwLength	4	00 00 00 05h もしくは 00 00 00 07h	このメッセージの abProtocolDataStructure データフィールドのサイズ。 データドメインはリトルエンドフォーマットで格納されます
5	bSlot	1	00-01h	このコマンドのスロット番号を識別します。 00h : PICC 01h : SAM
6	bSeq	1	00-FFh	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	bProtocolNum	1	00-01h	下記は指定されたプロトコルのデータ構造です。 00h = T=0 プロトコル構造 01h = T=1 プロトコル構造 下記の値を保留して将来に使用します： 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
8	abRFU	2	0000h	保留して将来使います。
10	abProtocolData Structure	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	bmFindexDindex	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	bmTCCKST0	1	00h、02h	B0 – 0b, B7-2 – 000000b B1 – 交換用 (b1=0 : 直接 ; b1=1 : 逆) 注 : CCID がこのビットを無視します。
12	bGuardTimeT0	1	00~FFh	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	bWaitingIntegerT0	1	00~FFh	T=0 の場合 WI が WWT を定義する時に使われます
14	bClockStop	1	00~03h	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	bmFindexDindex	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	BmTCKCKST1	1	10h, 11h, 12h, 13h	B7-2 – 000100b B0 – チェックサムタイプ (b0=0 : LRC ; b0=1 : CRC) B1 – 交換用 (b1=0 : 直接 ; b1=1 : 逆) 注 : CCID がこのビットを無視します。
12	BGuardTimeT1	1	00-FFh	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	BwaitingIntegerT1	1	00-9Fh	B7-4 = BWI 値 0-9 有効 B3-0 = CWI 値 0-Fh 有効
14	bClockStop	1	00-03h	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	bIFSC	1	00-FEh	交渉された IFSC の大きさ
16	bNadValue	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

5.2.4.7. RDR_to_PC_DataBlock

このコマンドは ACR1555U 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	bMessageType	1	80h	CCID がデータブロックを送信しています。
1	dwLength	4	00 00 00 00 ~ 00 00 01 E7h	このメッセージの abData データフィールドのサイズ。データドメインはリトルエンドフォーマットで格納されます
5	bSlot	1	-	Bulk-OUT メッセージ中の値と同じです。
6	bSeq	1	-	Bulk-OUT メッセージ中の値と同じです。
7	bStatus	1	-	付録 B 定義されたスロットビットステータスレジスタ
8	bError	1	-	付録 B 定義されたスロットビットエラーレジスタ
9	bChainParameter	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 データフィールドが継続に応答 APDU を転送し、APDU 応答を終了します。 03h - abData データデータフィールドが継続に応答 APDU を転送し、後ろ別のデータブロックが付いています。 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



オフセット	データフィールド	大きさ	数値	説明
10	abData	バイト 配列	-	このデータフィールドは CCID から返した データ

5.2.4.8. RDR_to_PC_SlotStatus

このコマンドは ACR1555U によって送信されて、PC_to_RDR_IccPowerOff と PC_to_RDR_GetSlotStatus 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	bMessageType	1	81h	-
1	dwLength	4	00 00 00 00h	メッセージの余分なバイトのサイズ。
5	bSlot	1	-	Bulk-OUT メッセージ中の値と同じです。
6	bSeq	1	-	Bulk-OUT メッセージ中の値と同じです。
7	bStatus	1	-	付録 B 定義のスロットステータスレジスタ
8	bError	1	-	付録 B 定義のスロットエラーレジスタ
9	bClockStatus	1	00~03h	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。

5.2.4.9. RDR_to_PC_Parameters

このコマンドは ACR1555U 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	bMessageType	1	82h	-
1	dwLength	4	00 00 00 05h もしくは 00 00 00 07h	このメッセージの abProtocolDataStructure データフィールドのサイズ。 データメインはリトルエンドフォーマットで格納されます
5	bSlot	1	-	Bulk-OUT メッセージ中の値と同じです。
6	bSeq	1	-	Bulk-OUT メッセージ中の値と同じです。

7	bStatus	1	-	付録 B 定義されたスロットビットステータスレジスタ
8	bError	1	-	付録 B 定義されたスロットビットエラーレジスタ
9	bProtocolNum	1	00-01h	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	abProtocolData Structure	バイト配列	-	プロトコルデータ構造が 5.2.4.6 概述をご参考ください

5.2.4.10. RDR_to_PC_Escape

このメッセージは ACR1555U によって送信されて、PC_to_RDR_Escape 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	bMessageType	1	83h	-
1	dwLength	4	00 00 00 00 ~ 00 00 00 FFh	このメッセージの abData データフィールドのサイズ。データドメインはリトルエンドフォーマットで格納されます
5	bSlot	1	-	Bulk-OUT メッセージ中の値と同じです。
6	bSeq	1	-	Bulk-OUT メッセージ中の値と同じです。
7	bStatus	1	-	付録 B 定義されたスロットビットステータスレジスタ
8	bError	1	-	付録 B 定義されたスロットビットエラーレジスタ
9	bChainParameter	1	00h	RFU

オフセット	データフィールド	大きさ	数値	説明
10	abData	バイト 配列	-	C C ID から送信されたデータ

5.2.4.11. RDR_to_PC_Error

デバイスが間違ったコマンドを受信した場合、このメッセージによりエラー情報を返します。

オフセット	データフィールド	大きさ	数値	説明
0	bMessageType	1	53h	-
1	dwLength	4	00 00 00 00h	メッセージの余分なバイトのサイズ。
5	bSlot	1	-	Bulk-OUT メッセージ中の値と同じです。
6	bSeq	1	-	Bulk-OUT メッセージ中の値と同じです。
7	bStatus	1	-	付録 B 定義のスロットステータスレジスタ
8	bErrorCode	1	-	01h = チェックサムエラー 02h = タイムアウト 03h = コマンドエラー 04h = 許可とれていません 05h = 未定義のエラー 06h = 受信データにエラーある 07h = 受信データの長さにエラーある 08h = 認定再試行回数超過エラー
9	abRFU	1	00h	RFU

5.2.4.12. RDR_to_PC_NotifySlotChange

このメッセージはカードの状態を通知するためにリーダ/ライタによって発行されます。

オフセット	データフィールド	大きさ	数値	説明
0	bMessageType	1	50h	-
1	bmSlotICCState	1	-	状態 : 02h = PICC カードなし 03h = PICC カードある



5.2.4.13. RDR_to_PC_Sleep

このメッセージはスリープモードの状態を通知するためにリーダ/ライタによって発行されます。

オフセット	データフィールド	大きさ	数値	説明
0	bMessageType	1	52h	-
1	bParam	1	00h	RFU

5.2.5. 認証

強制暗号化が有効になっている場合、ACR1555U に機密データをロードする前に、リーダライタ内部のセキュリティデータを変更する権限を持つために、データ処理サーバは ACR1555U の認証を取得する必要があります。ACR1555U は相互認証を採用します。

良く説明するために、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）

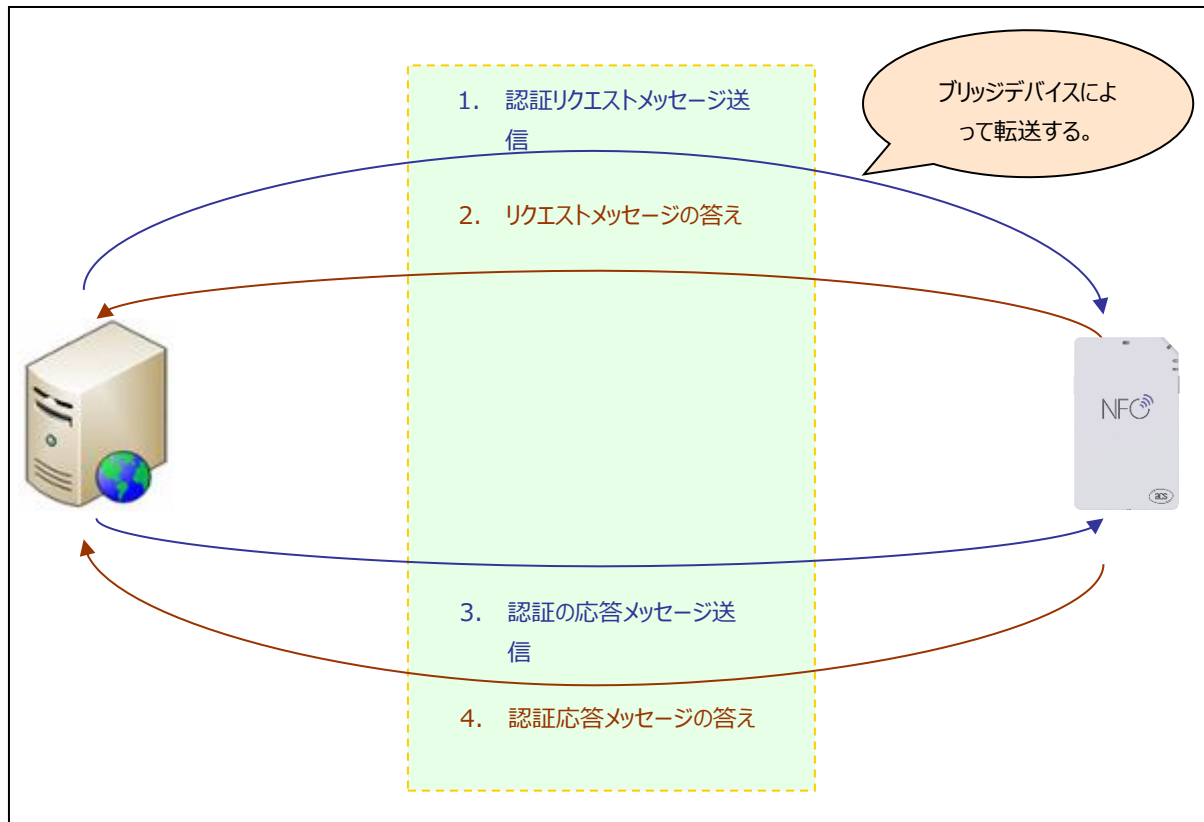


図5：認証手順

認証が成功すると、ACR1555U とデータサーバが 16 バイトのセッションキーを生成する。

デフォルト顧客マスターキー（16 進法で **41 43 52 31 35 35 35 55 2D 41 31 20 41 75 74 68**）

重要提示：

間違った認証キーを入力すると、カードリーダーは永久に使用できなくなります。必ず正しいキーを入力してください。送信する前に入力内容をよく確認してください。

詳細については、『開発者ガイド：双方向認証とセキュリティデータ交換』を参照してください。ACS 営業担当者に連絡することもできます。

5.3. PCSC API

このセクションでは、アプリケーションプログラミング用の PCSC API について説明します。これらの API の詳細について、Microsoft MSDN ライブラリまたは PCSC ワークグループ仕様サイトを参照してください。

5.3.1. SCardEstablishContext

SCardEstablishContext 関数はデータベース操作を実行するリソースマネージャのコンテキストを確立するためです。

参照してください：<http://msdn.microsoft.com/en-us/library/windows/desktop/aa379479%28v=vs.85%29.aspx>

他の PCSC アクションを実行する前に、この関数を実行してください。

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT hContext;
int retCode;
void main ()
{
    // To establish the resource manager context and assign it to "hContext"
    retCode = SCardEstablishContext(SCARD_SCOPE_USER,
                                    NULL,
                                    NULL,
                                    &hContext);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Establishing resource manager context failed
    }
    else
    {
        // Establishing resource manager context successful
        // Further PCSC operation can be performed
    }
}
```

5.3.2. SCardListReaders

SCardListReaders 関数を使用して、システム内に指定されたカードリーダーグループコレクション内のカードリーダーリストを取得します（重複項目排除）。

呼び出し側がカードリーダーグループのリストを提供して、関数が指定されたグループ内のカードリーダー名のリストを返す。認識できないグループの名前は無視されます。この関数が現在システムに利用できるグループ中のリーダーだけ返されます。

参照してください：

<http://msdn.microsoft.com/en-us/library/windows/desktop/aa379793%28v=vs.85%29.aspx>

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT hContext; // Resource manager context
int retCode;
char readerName [256]; // List reader name

void main ()
{
    To establish the resource manager context and assign to "hContext"
    retCode = SCardEstablishContext(SCARD_SCOPE_USER,
                                    NULL,
                                    NULL,
                                    &hContext);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Establishing resource manager context failed
    }
    else
    {
        // Establishing resource manager context successful
        // List the available reader which can be used in the system
        retCode = SCardListReaders (hContext,
                                    NULL,
                                    readerName,
                                    &size);
        if (retCode != SCARD_S_SUCCESS)
        {
            // Listing reader fail
        }
        if (readerName == NULL)
        {
            // No reader available
        }
        else
        {
            // Reader listed
        }
    }
}
```

例：

5.3.3. SCardConnect

SCardConnect 関数は特定のエクスポーラコンテキストを使用して、アプリケーションと特定のカードリーダーに含まれるスマートカードとの間の接続を確立します。特定のリーダー中にはカードがない場合、エラーメッセージが返されます。

参照してください：

<http://msdn.microsoft.com/en-us/library/windows/desktop/aa379473%28v=vs.85%29.aspx>

例：

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT    hContext;           // Resource manager context
SCARDHANDLE      hCard;             // Card context handle
unsigned long    dwActProtocol;     // Establish active protocol
int              retCode;
char             readerName [256];  // List reader name
char             rName [256];      // Reader name for connection

void main ()
{
    ...
    if (readerName == NULL)
    {
        // No reader available
    }
    else
    {
        // Reader listed
        rName = "ACS ACR1555 1S CL Reader PICC 0"; // Depends on what
                                                    // reader be used
                                                    // Should connect to
                                                    // PICC interface

        retCode = SCardConnect(hContext,
                                rName,
                                SCARD_SHARE_SHARED,
                                SCARD_PROTOCOL_T0,
                                &hCard,
                                &dwActProtocol);
        if (retCode != SCARD_S_SUCCESS)
        {
            // Connection failed (May be because of incorrect reader
            // name, or no card was detected)
        }
        else
        {
            // Connection successful
        }
    }
}
```

5.3.4. SCardControl

SCardControl 関数は、SCardConnect 関数を正常に呼び出した後、SCardDisconnect 関数を正常に呼び出す前にいつでも呼び出すことができるカードリーダーの直接制御を提供します。リーダーの状態に対する影響は、制御コードに依存しています。

参照してください：

<http://msdn.microsoft.com/en-us/library/windows/desktop/aa379474%28v=vs.85%29.aspx>

注：**Escape コマンド**中のコマンドはこの API を使用して送信する必要があります。

例：

```
#define SCARD_SCOPE_USER    0

#define EscapeCommand 0x310000 + 3500*4
SCARDCONTEXT      hContext;           // Resource manager context
SCARDHANDLE        hCard;             // Card context handle
unsigned long      dwActProtocol;     // Established active protocol
int               retCode;
char               readerName [256];  // Lists reader name
char               rName [256];       // Reader name for connection
BYTE               SendBuff[262];     // APDU command buffer
BYTE               RecvBuff[262];     // APDU response buffer
BYTE               FWVersion [20];    // For storing firmware
                                   version message
BYTE               ResponseData[50];  // For storing card response
DWORD              SendLen,           // APDU command length
                   RecvLen;           // APDU response length

void main ()
{
    ...
    rName = "ACS ACR1555 1S CL Reader PICC 0"; // Depends on what
                                                reader will be used
                                                // Should connect to
                                                PICC interface

    retCode = SCardConnect(hContext,
        rName,
        SCARD_SHARE_DIRECT,
        SCARD_PROTOCOL_T0| SCARD_PROTOCOL_T1,
        &hCard,
        &dwActProtocol);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Connection failed (may be because of incorrect reader
        name, or no card was detected)
    }
    else
    {
        // Connection successful
        RecvLen = 262;
        // Get firmware version
        SendBuff[0] = 0xE0;
        SendBuff[1] = 0x00;
        SendBuff[2] = 0x00;
        SendBuff[3] = 0x18;
        SendBuff[4] = 0x00;
    }
}
```



```
SendLen = 5;
retCode = SCardControl ( hCard,
    EscapeCommand,
    SendBuff,
    SendLen,
    RecvBuff,
    RecvLen,
    &RecvLen);
if (retCode != SCARD_S_SUCCESS)
{
    // APDU sending failed
    return;
}
else
{
    // APDU sending successful
    // The RecvBuff stores the firmware version message.
    for (int i=0;i< RecvLen-5;i++)
    {
        FWVersion[i] = RecvBuff [5+i];
    }
}
// Connection successful
RecvLen = 262;

// Turn Green LED on, turn Red LED off
SendBuff[0] = 0xE0;
SendBuff[1] = 0x00;
SendBuff[2] = 0x00;
SendBuff[3] = 0x29;
SendBuff[4] = 0x01;
SendBuff[5] = 0x02; // Green LED On, Red LED off
SendLen = 6;
retCode = SCardControl ( hCard,
    EscapeCommand,
    SendBuff,
    SendLen,
    RecvBuff,
    RecvLen,
    &RecvLen);
if (retCode != SCARD_S_SUCCESS)
{
    // APDU sending failed
    return;
}
else
{
    // APDU sending success
}
```

5.3.5. SCardTransmit

SCardTransmit 関数はサービス請求をスマートカードに送信するために、またはスマートカードから返されるデータを受信するために使われます。

参照してください：

<http://msdn.microsoft.com/en-us/library/windows/desktop/aa379804%28v=vs.85%29.aspx>

注： この API で APDU コマンド（即ち：接続を確立されたカードに送信するコマンド PICC の PCSC 私有 APDU (独自の拡張機能付き) と PICC の専属の私有 APDU を送信します。

例：

```
#define SCARD_SCOPE_USER      0

SCARDCONTEXT      hContext;          // Resource manager context
SCARDHANDLE        hCard;            // Card context handle
unsigned long      dwActProtocol;     // Established active protocol
int                retCode;
char               readerName [256];  // List reader name
char               rName [256];      // Reader name for connect
BYTE               SendBuff[262];     // APDU command buffer
BYTE               RecvBuff[262];     // APDU response buffer
BYTE               CardID [8];        // For storing the FeliCa IDM/
                                      MIFARE UID
BYTE               ResponseData[50];  // For storing card response
DWORD              SendLen,           // APDU command length
                  RecvLen;            // APDU response length
SCARD_IO_REQUEST   ioRequest;

void main ()
{
    ...
    rName = "ACS ACR1555 1S CL Reader PICC 0"; // Depends on what
                                                reader should be used
                                                // Should connect to PICC
                                                interface

    retCode = SCardConnect(hContext,
                           rName,
                           SCARD_SHARE_SHARED,
                           SCARD_PROTOCOL_T0,
                           &hCard,
                           &dwActProtocol);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Connection failed (May be because of incorrect reader
        // name, or no card was detected)
    }
    else
    {
        // Connection successful
        ioRequest.dwProtocol = dwActProtocol;
        ioRequest.cbPciLength = sizeof(SCARD_IO_REQUEST);
        RecvLen = 262;
    }
}
```



```
// Get MIFARE UID/ FeliCa IDM
SendBuff[0] = 0xFF;
SendBuff[1] = 0xCA;
SendBuff[2] = 0x00;
SendBuff[3] = 0x00;
SendBuff[4] = 0x00;
SendLen = 5;
retCode = SCardTransmit( hCard,
                        &ioRequest,
                        SendBuff,
                        SendLen,
                        NULL,
                        RecvBuff,
                        &RecvLen);

if (retCode != SCARD_S_SUCCESS)
{
    // APDU sending failed
    return;
}
else
{
    // APDU sending successful
    // The RecvBuff stores the IDM for FeliCa / the UID for
    MIFARE.
    // Copy the content for further FeliCa access
    for (int i=0;i< RecvLen-2;i++)
    {
        CardID [i] = RecvBuff[i];
    }
}
```

5.3.6. SCardDisconnect

SCardDisconnect 関数は前に確立されたアプリケーションとターゲットリーダー間の接続を終了するためです。。

参照してください：

<http://msdn.microsoft.com/en-us/library/windows/desktop/aa379475%28v=vs.85%29.aspx>

この関数が PCSC 操作をエンドするために使用されます。

例：

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT      hContext;          // Resource manager context
SCARDHANDLE        hCard;            // Card context handle
unsigned long      dwActProtocol;    // Established active protocol
int                retCode;

void main ()
{
    ...
    // Connection successful
    ...
    retCode = SCardDisconnect(hCard, SCARD_RESET_CARD);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Disconnection failed
    }
    else
    {
        // Disconnection successful
    }
}
}
```

5.3.7. APDU の流れ

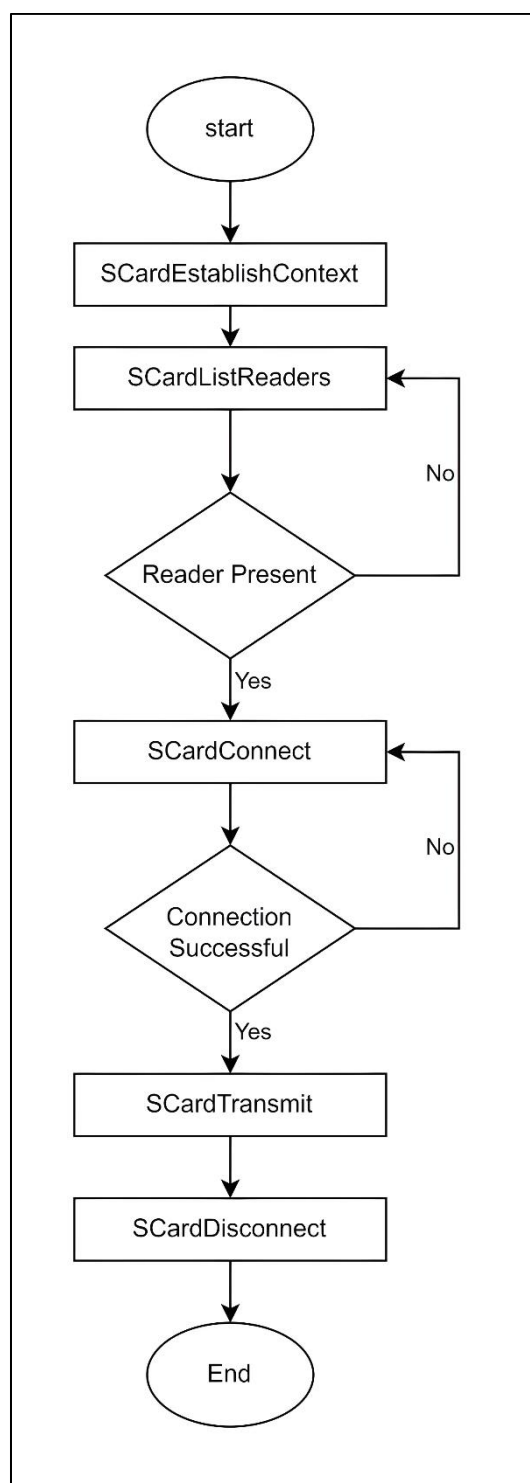


図6 : ACR1555U の APDU 流れ

5.3.8. 直接なコマンドの流れ

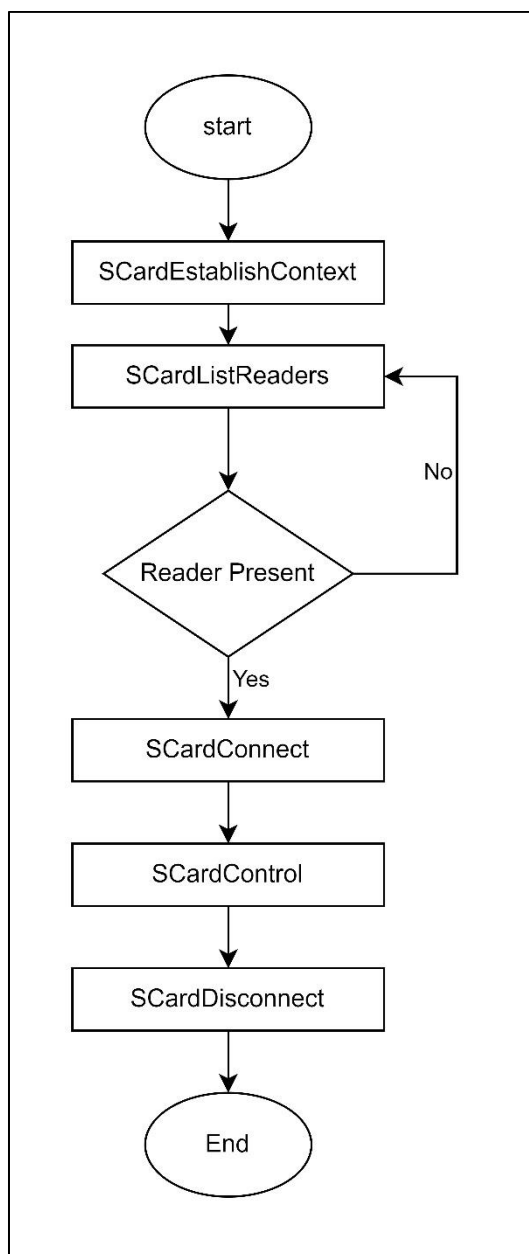


図7 : ACR1555U 直接コマンドの流れ

5.4. 接触式スマートカードプロトコル

5.4.1. ACOS6-SAM カードコマンド

このセクションでは、SAM 専用のコマンドを説明します。CCID ホストが PCSC API 中の SCardTransmit () に対応する CCID メッセージ PC _ to _ RDR _ XfrBlock を使用して、カードリーダーにカードアドミニストレーションコマンドまたは APDU を送信することができます。

注：ACOS 6-SAM コマンドのすべての情報とアプリケーションシーンについては、ACS 営業担当者に問い合わせ、ACOS 6-SAM リファレンスマニュアルを請求してください。

5.4.1.1. キー生成 (Generate Key)

このコマンドは、顧客カードのシリアル番号などの偏差データを用いて分散鍵を生成し、ACOS 3/6 または他のカードに導入することで、顧客カード発行の目的を満たす。

APDU	説明
CLA	80h
INS	88h
P1	00h 8 バイトのキーを生成する
	01h 16 バイトのキーを生成する
	02h 24 バイトのキーを生成する
P2	導出鍵を生成するためのマスター鍵の鍵インデックス
P3	08h
データ	データ入力：

特定の応答ステータスバイト

SW1 SW2	説明
69 86h	DF 未選択
6A 86h	P1 もしくは P2 無効
67 00h	P3 が正しくない、08h でなければならない
6A 83h	指定された鍵レコードが EF 2 には見つかりません
69 81h	無効な EF2 (レコードサイズ、ファイルタイプなど)
6A 88h	EF2 が見つかりません
62 83h	現在の DF はロックされている; EF2 がロックされている
69 83h	使用カウンタはゼロです。
69 82h	セキュリティ条件が満たされていない



SW1	SW2	説明
6A	87h	指定されたマスターキーは 3-DES 暗号化サポートしない
61	08h	コマンドが完了、結果を得るために GET REPOSE を送信する

5.4.1.2. キーデータ分散化（もしくはロード）（Diversify (or Load) Key Data）

このコマンドは、鍵分散と鍵ロードによって SAM カードに暗号化操作を実行する準備をさせます。コマンドデータ入力として、シリアル番号と CBC 初期ベクトルを取ります。

APDU		説明							
CLA	80h								
INS	72h								
P1	b7	b6	b5	b4	b3	b2	b1	b0	説明
	-	0	0	0	0	0	0	1	パスワード(Sc)
	-	0	0	0	0	0	1	0	アカウントキー(K _{ACCT})
	-	0	0	0	0	0	1	1	エンドキー
	-	0	0	0	0	1	0	0	カードキー
	-	0	0	0	0	1	0	1	キーを一括に暗号化する（分散しない）
	-	0	0	0	0	1	1	0	初期ベクトル
	0	-	-	-	-	-	-	-	16 バイトのキー
	1	-	-	-	-	-	-	-	24 バイトのキー
マスターキーの索引：									
P2	Bit7:	1 =現在 EF2 中のローカルキー ； 0 =グローバルキー EF2							
	Bit6-Bit5:	00b - RFU							
	Bit4-Bit0:	キーインデックス							
P3	P1 = 1～4 の場合、P3 = 8/16（アルゴリズムが AES の場合、P3 = 8/16）								
	P1 = 5 の場合、P3 = 0								
	P1 = 6 の場合、 P3 = 8（マスターキーの Algo は DES / 3DES / 3KDES） P3 = 16（マスターキーの Algo は AES）								
データ	P1 = 1-4 の場合、クライアントカードのシリアル番号（algo が AES の場合、データはクライアントカードのシリアル番号またはクライアントカードのシリアル番号に「0000000000000000」が付加されます） P1 = 5 の場合、コマンドデータはありません。 P1 = 6 の場合、DES / 3DES / 3KDES / AES CBC の初期ベクトル。								

特定の応答ステータスバイト

SW1 SW2	説明
69 86h	DF 未選択
6A 86h	間違った P1、P1 は 1～6 でなければなりません
67 00h	間違った P3、P3 は 8（または 0）
62 83h	現在の DF がロックされているか、または EF2 がロックされている
69 82h	セキュリティ条件が満たされていない
6A 88h	EF2 が見つかりません
6A 83h	EF2 に指定マスターキーが見つかりません
69 81h	無効な EF2（FDB、MRL など、一貫性がない）
6A 87h	指定された KEY 認証できない
69 83h	参照されたキーがロックされています
90 00h	ターゲットキーが生成され、SAM メモリに準備されている

5.4.1.3. 暗号化 (Encrypt)

このコマンドは、DES または 3DES を使用して次のいずれかの方法でデータを暗号化します。

1. ACOS3 / 6、DESFire®、DESFire® EV1 または MIFARE Plus カードとの相互認証手順によって作成されたセッションキー。
2. 分散化した鍵（パスワード）。
3. キーを一括に暗号化する。
4. セッション鍵で分散化したパスワードを暗号化します。
5. 非 SM コマンドを指定して、ACOS3 セキュア・メッセージング・コマンドを作成します。

APDU	説明
CLA	80h
INS	74h



APDU		説明								
		b7	b6	b5	b4	b3	b2	b1	b0	説明
P1	-	0	0	0	0	0	0	0	-	ECB モード
	-	0	0	0	0	0	0	1	-	CBC モード
	-	0	0	0	0	0	1	0	-	小売 MAC モード
	-	0	0	0	0	0	1	1	-	MAC モード
	-	0	0	0	0	1	0	0	-	ACOS3 SM コマンド用意する
	-	1	0	0	0	1	0	1	-	MIFARE DESFire 暗号化
	-	1	0	0	0	1	1	0	-	MIFARE DESFire EV1 暗号化
	-	0	0	0	0	1	1	1	-	CMAC
	-	0	1	0	0	0	0	0		MIFARE Plus コマンド
	-	0	1	0	0	0	0	1		MIFARE Plus 応答
	0	-	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	-	0	3K DES
	1	-	-	-	-	-	-	-	1	AES
-	-	-	-	-	-	-	-	-	他のすべての値 - RFU	
P 2 は、Load Key 機能を使用して SAM セットに分散された鍵を表します。										
P2	1	セッションキー Ks でデータを暗号化する								
	2	多様なキー Sc でデータを暗号化する								
	3	一括暗号鍵でデータを暗号化する								
	0	ENC (Sc、Ks) を返す								
P1.b3 = 1 または b5 = 1 の場合、P2 は 1 でなければなりません										
P2 = 0h の場合、P1 は 0 または 1 のいずれかになります										
P3 < 128										
P1 のビット 3 が 1 に等しくなく、P1 のビット 5 が 1 に等しくない場合										
P3	-	P2 = 1-3 の場合、8 (DES / 3DES / 3KDES) の倍数または 16 バイト (AES) の 128 バイト								
	-	P2 = 0 の場合、0								

APDU 説明

プレーンテキスト

P2 b6 = 1 の場合、データのフォーマットが次のようになります：

- プレーンテキストデータの長さ
- DESFire カードのコマンドとヘッダーの長さ
- DESFire カードのコマンドとヘッダー
- プレーンテキスト

P1 = A1h、の場合、暗号化は Plus コマンド用です

データ

- MFP コマンドが値操作コマンドの場合、データのフォーマットは Command Code (1 バイト) + BlockNum (2/4 バイト) + Value (4 バイト) でなければなりません。
- MFP コマンドが Proximity Check の場合、データフォーマットは Command Code (1 バイト) + PPS1 (1 バイト) でなければなりません。
- MFP コマンドが読み取りの場合、データフォーマットは Command code (1 バイト) + BlockNum (2 バイト)
- MFP コマンドが書き込みである場合、データフォーマットは Command code (1 バイト) + BlockNum (2 バイト) + plaintext

P1 = A3h、

- ICC によって返されたデータ (SC コードは含まず、RMAC が存在する場合は RMAC を含まない)

特定の応答ステータスバイト

SW1 SW2	説明
69 86h	DF 未選択
6A 86h	P1 もしくは P2 無効
67 00h	P3 正しくない
6A 83h	ACOS ターゲットキーが準備されていません (キーを生成するために Diversify を使用してください)
61 XX	暗号化が行われ、結果を得るために GET RESPONSE を使用する

5.4.1.4. 復号化 (Decrypt)

このコマンドは、DES または 3DES または AES を使用して次のいずれかを使用してデータを復号化するために使用されます。

1. ACOS3/6、MIFARE DESFire、MIFARE DESFire EV1またはMIFARE Plusカードとの相互認証手順によって作成されたセッションキー。
2. 多様化した鍵（パスワード）。
3. Aバルク暗号化キー。
4. セッション鍵で多様化したパスワードを解読化します。
5. ACOS3の安全なメッセージングの応答を確認し、解読する。

ACOS3 の安全なメッセージングの応答を確認し、解読する。

APDU 説明									
CLA	80h								
INS	76h								
		b7	b6	b5	b4	b3	b2	b1	b0 説明
		-	0	0	0	0	0	0	- ECB モード
		-	0	0	0	0	0	1	- CBC モード
		-	0	0	0	1	0	0	- ACOS3 の安全なメッセージングの応答を確認し、解読する。
P1		-	1	0	0	1	0	1	- MIFARE DESFire で復号化
		-	1	0	0	1	1	0	- MIFARE DESFire EV1 で復号化
		-	0	1	0	0	1	0	- MIFARE Plus 復号化
		0	-	-	-	-	-	-	0 3DES
		0	-	-	-	-	-	-	1 DES
		1	-	-	-	-	-	-	0 3K DES
		1	-	-	-	-	-	-	1 AES
		0	0	0	0	-	-	-	- 他のすべての値 - RFU
P 2 は、Load Key 機能を使用して SAM セットに分散された鍵を表します。									
P2		1 - セッションキー-Ks でデータを復号化する							
		2 - 多様なキー-Sc でデータを復号化する							
		3 - 一括暗号鍵でデータを復号化する							
		0 - DEC (Sc、Ks) を返す							

APDU	説明
P3	P3 < 128 P1 = A5h の場合、P3 = 16/32/48 P1 のビット 3 が 1 に等しくない場合 - P2 = 1-3 の場合、8 (DES / 3DES / 3KDES) の倍数または 16 バイト (AES) の 128 バイト - P2 = 0 の場合、0
	暗号文 P1 = A5h の場合、データが暗号化されたテキスト P2 b6 = 1 の場合、データのフォーマットが次のようになります : データ • Plain テキストデータの長さ (不明の場合は 00 を使用) • DESFire カードのコマンドとヘッダーの長さ • DESFire カードのコマンドとヘッダー • 暗号化されたテキスト

特定の応答ステータスバイト

SW1 SW2	説明
69 86h	DF 未選択
6A 86h	P1 もしくは P2 無効
67 00h	P3 正しくない
6A 83h	ACOS ターゲットキーが準備されていません (キーを生成するために Diversify を使用してください)
61 XX	解読化が行われ、結果を得るために GET RESPONSE を使用する

5.4.1.5. 認証準備 (Prepare Authentication)

このコマンドは、ACOS 3/6、MIFARE Ultralight C/MIFARE DESFire/MIFARE Plus カードに対して SAM カード（端末として）を認証するために使用されます。

APDU	説明
CLA	80h
INS	78h
P1	00h – 3DES 01h – DES 02h – 3KDES (MIFARE DESFire EV1/ACOS3) 03h – AES (MIFARE DESFire EV1/MIFARE Plus/ACOS3) 80h – 3DES (MIFARE DESFire 認証のみ) 81h – DES (MIFARE DESFire 認証のみ) 他 – RFU
P2	0h - ACOS3 / 6 が認証を返すことを確認する 01h - MIFARE 超軽量 C / DESFire 認証（多様化）ターミナルキー 05h - MIFARE Ultralight C / DESFire は一括暗号鍵で認証します 02h - MIFARE Plus 認証 SL1 から SL3 の最初の認証 03h - MIFARE Plus 認証。SL1 から SL2 への認証。 04h - MIFARE Plus 認証。SL2 から SL3 への認証に続きます。
P3	8 – (P1 = 00h, 01h, 02h, 80h, 81h) 16 – (P1 = 03h)
データ	カードチャレンジデータ

特定の応答ステータスバイト

SW1 SW2	説明
69 86h	DF 未選択
6A 86h	P1 もしくは P2 無効
67 00h	P3 が正しくない、08h でなければならない
6A 83h	ACOS キー（KT または KC）が準備できていない（Diversify でこのキーを生成してください）
69 82h	セキュリティ条件が満たされていない
61 10h	コマンドが完了、結果を得るために GET RESPONSE を送信する

5.4.1.6. 照合認証 (Verify Authentication)

このコマンドは、ACOS 3/6、MIFARE Ultralight C、MIFARE DESFire/MIFARE DESFire EV1 または MIFARE Plus カードが端末に対する正当性を検証するために使用され、内部でプロセスキーKs も生成されます。

APDU	説明
CLA	80h
INS	7Ah
P1	00h – 3DES (P2 = 0) 01h – DES (P2 = 0) 02h – 3KDES (P2 = 0, ACOS3) 03h – AES (P2 = 0, ACOS3) 他 – RFU
P2	00h - ACOS3 /6 認証で返されたメッセージを検証する 01h - MIFARE Ultralight C®/ DESFire®/ DESFire® EV1 認証で返されたメッセージを検証する 02h - MIFARE Plus の認証状況を確認する
P3	08h - (P2 = 0、P2 = 1、セッションキーは DES / 3DES) 16h - (P2 = 1、セッションキーは 3KDES / AES 使用) 16h - (P2 = 02、MIFARE Plus がリターンデータ ek (RndA')) 32h - (P2 = 02、MIFARE Plus がリターンデータ ek (TI + PICCcap2 + PCDcap2))
データ	ACOS 3/6 : DES (Ks, RND _T) MIFARE DESFire/ DESFire EV1 がリターンデータ : ek(RndA') MIFARE Plus がリターンデータ ek (RndA') または ek (TI + PICCcap2 + PCDcap2)

特定の応答ステータスバイト

SW1 SW2	説明
69 86h	DF 未選択
6A 86h	P1 もしくは P2 無効
67 00h	P3 が正しくない、08h でなければならない
6A 83h	ACOS-SAM セッションキーまたは RNDT は準備ができていません。PREPARE AUTHENTICATION を使用してこれらのキーを作成します。
69 82h	データ正しくない



SW1	SW2	説明
90	00h	データ正しくて ACOS 相互認証成功

5.4.1.7. ACOS クエリアカウントチェック (Verify ACOS Inquire Account)

このコマンドは、ACOS3 / 6 カードの Inquire Account purse コマンドを確認するために使用します。それは、ACOS3 / 6 によって返された MAC チェックサムが、SAM の多様化された鍵で正しいことを検証する。

APDU	説明								
CLA	80h								
INS	7Ch								
P1	b7	b6	b5	b4	b3	b2	b1	b0	説明
	-	0	0	0	0	-	0	-	ACOS INQ_AUT 無効
	-	0	0	0	0	-	1	-	ACOS INQ_AUT 有効
	-	0	0	0	0	0	-	-	ACOS INQ_ACC_MAC 無効
	-	0	0	0	0	1	-	-	ACOS INQ_ACC_MAC 無効
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES（ACOS3 のみ）
	1	-	-	-	-	-	-	1	AES（ACOS3 のみ）
P2	0h								
P3	1Dh								
データ	クライアントの ACOS カードの INQUIRE ACCOUNT によって返されるデータブロック。以下を参照してください。								

特定の応答ステータスバイト

SW1 SW2	説明
69 86h	DF 未選択
6A 86h	P1 もしくは P2 無効
67 00h	P3 正しくない
6A 83h	ACOS キー K _S または K _{ACCT} は準備ができていません。K _{ACCT} 生成するには DIVERSIFY コマンドを使用します。該当する場合は、「Prepare Authentication」を介して K _S を生成します。
6F 00h	データブロックの MAC が正しくありません
90 00h	データブロックの MAC が正しい

5.4.1.8. ACOS アカウント取引準備 (Prepare ACOS Account Transaction)

ACOS3 / 6 クレジット/デビットコマンドを作成するには、ACOS3 / 6 が検証するために MAC を計算する必要があります。

APDU		説明							
CLA	80h								
INS	7Eh								
	b7	b6	b5	b4	b3	b2	b1	b0	説明
	-	0	0	0	0	0	0	-	ACOS TRNS_AUT 無効
	-	0	0	0	0	0	1	-	ACOS TRNS_AUT 有効
P1	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES (ACOS3 のみ)
	1	-	-	-	-	-	-	1	AES (ACOS3 のみ)
P2	E2h: クレジット								
	E6h: デビット								
P3	0Dh								
データ	データブロック								

特定の応答ステータスバイト

SW1 SW2	説明
69 86h	DF 未選択
6A 86h	P1 もしくは P2 無効
67 00h	P3 が正しくない、必ず 0Dh
6A 83h	ACOS キー-K _S または K _{ACCT} は準備ができていません。K _{ACCT} 生成するには DIVERSIFY コマンドを使用します。該当する場合は、「Prepare Authentication」を介して K _S を生成します。
61 0Bh	コマンドが完了、結果を得るために GET REPONSE を送信する

5.4.1.9. デビット証明書認証 (Verify Debit Certificate)

ACOS3 / 6 の場合、DEBIT コマンドに P1 = 1 が指定されている場合は、借方の証明書が返されます。デビット証明書は、ACOS3 の応答をこのコマンドの結果と比較することによって確認できます。

APDU 説明									
CLA	80h								
INS	70h								
P1	b7	b6	b5	b4	b3	b2	b1	b0	説明
	-	0	0	0	0	0	0	-	ACOS TRNS_AUT 無効
	-	0	0	0	0	0	1	-	ACOS TRNS_AUT 有効
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES (ACOS3 のみ)
	1	-	-	-	-	-	-	1	AES (ACOS3 のみ)
	P2	0h							
P3	14h								
データ	データブロック								

特定の応答ステータスバイト

SW1 SW2	説明
69 86h	DF 未選択
6A 86h	P1 もしくは P2 無効
67 00h	P3 が正しくない、14h でなければならない
6A 83h	ACOS キー-Ks または K _{ACCT} は準備ができていません。K _{ACCT} 生成するには DIVERSIFY コマンドを使用します。該当する場合は、「Prepare Authentication」を介して Ks を生成します。
69 82h	セキュリティ条件が満たされていない
6F 00h	DEBIT CERTIFICATE 無効
90 00h	成功, DEBIT CERTIFICATE 有効

5.4.1.10. キー取得 (Get Key)

鍵を取るコマンドにより、鍵を現在の SAM の鍵ファイル (SFI=02 h) から別の ACOS 6/ACOS 6-SAM カードに安全に注入することができ、このプロセスは、鍵分散を介さなくても実現できます。これを使用すると、注入される鍵が暗号化およびメッセージ認証コードによって保護されます。

またこのコマンドが分散化により、現在の SAM のキーファイル（SFI = 02h）から ACOS7/10、MIFARE DESFire、MIFARE DESFire EV1 または MIFARE Plus カードに安全に注入することができます。これを使用すると、注入される鍵が暗号化およびメッセージ認証コードによって保護されます。

カードヘッダーブロック（ACOS6-SAM リファレンスマニュアルのセクション 3.2）の特殊機能フラグ（キー注入専用フラグ）のビット 7 が設定され、キーファイルが有効になっている場合、キーのロードまたは変更には Get キーを使用する必要があります。Bit 7 が設定されると、鍵ファイルがアクティブになると、Read Record コマンドの使用はいずれの場合も無効になります。

このコマンドを実行する前に、**相互認証**の相互認証プロセス（ACOS6-SAM リファレンスマニュアルのセクション 5.3）または MIFARE Plus/MIFARE DESFire 相互認証プロセスで、ターゲットカードとのセッション鍵がすでに確立されています。

注：GET KEY コマンドはキーデータのみを取得できます。

APDU		説明		
CLA	80h			
INS	CAh			
ACOS カードがキーを取得、リセットする用				
00h		応答データは MSAM のキーです		
01h		応答データは 16 バイトの Diversify キーです		
02h		応答データは 24 バイトの Diversify キーです		
03h		応答データは MIFARE Plus カードの Change Key コマンドです		
DESFire カードのキー取得変更キー、DESFire / DESFire EV1 Change キーの応答データ				
		カードタイプ	キー番号の認証とキー番号の変更*	キー長さ
P1	80h	MIFARE DESFire	MIFARE®DESFire の中では異なる	16 バイト
	81h	MIFARE DESFire EV1	MIFARE DESFire EV1 カードの中では異なる	16 バイト
	82h	MIFARE DESFire EV1	MIFARE DESFire EV1 カードの中では異なる	24 バイト
	88h	MIFARE DESFire	MIFARE DESFire の中では同じ	16 バイト
	89h	MIFARE DESFire EV1	MIFARE DESFire EV1 カードの中では同じ	16 バイト
	8Ah	MIFARE DESFire EV1	MIFARE DESFire EV1 カードの中では同じ	24 バイト
P2	SAM のキー-ID（変更のための新しいキー）			

APDU	説明
P3	P1 = 00h, P3 = 08h
	P1 = 02h, P3 = 10h
	P1 = 03h, P3 = 0h
	P1 = 80/81/82/88/89/8Ah : P3 = 0Bh
データ	P1 = 00h の場合、コマンドデータは RND _{Target}
	P1 = 01 / 02h の場合、コマンドデータは RND _{Target} + ターゲットカードのシリアル（またはロット）番号です
	P1 = 03h の場合 P1 = 03h
	- ターゲットカードのシリアル番号（8 バイト）
	- ライトコマンド（A0 または A1）（1 バイト）
	- BNr（2 バイト）
	P1 = 80/81/82/88/89/8Ah :
	- ターゲットカードのシリアル番号（8 バイト）
	- 元の鍵 ID（元の鍵が格納されている SAM カードの鍵、00 = DESFire のデフォルト鍵 - カード）
	- 鍵番号（DESFire カード鍵番号）
	- Key Version（DESFire カードのキーバージョン、使用されない場合、値= 00）

*このリストは、リストされているカードが違う Change Key と Authenticate Key を持っているか、または 2 つの鍵が同じ値を使用しているかを示しています。

特定の応答ステータスバイト

SW1 SW2	説明
69 85h	SAM セッションキーが準備完了していません
62 83h	現在の DF がブロックされているか、またはターゲット EF がブロックされている
69 86h	DF 未選択
69 81h	キーファイルのファイルタイプが間違っています。内部線形変数ファイル
69 82h	ターゲットファイルのヘッダブロックのチェックサムが正しくないか、セキュリティ条件が満たされていません
6A 86h	P1 もしくは P2 無効
67 00h	P3 正しくない
6A 83h	ターゲットキーが準備されていないか、キーの長さが 16 未満です
61 1Ch	コマンド成功、GET RESPONSE で結果を取得

5.5. 非接触スマートカード プロトコル

5.5.1. ATR の生成

リーダーが PICC を検出すると、PICC を識別するために、ATR が PCSC ドライバに送られます。

5.5.1.1. ATR フォーマット (ISO14443-3-3 PICC に適用)

バイト	数値	標記	説明
0	3Bh	最初のヘッダー	
1	8Nh	T0	高いニブル8の意味は：TA1、TB1とTC1がなくて、TD1だけが続いている。 下位ニブル N は歴史的なバイトの数です (HistByte 0 - HistByte N-1)
2	80h	TD1	高いニブル8の意味は：TA2、TB2とTC2がなくて、TD2だけが続いている。 下位ニブル 0 の意味は T=0
3	01h	TD2	高いニブル0の意味は：TA3、TB3、TC3およびTD3が全部続いていない。 下位ニブル 1 の意味は T=1
4 ~ 3+N	80h	T1	カテゴリインジケータバイトは、80のステータスインジケータが任意の COMPACT-TLVデータオブジェクトに存在するかもしれない意味です
	4Fh	Tk	アプリケーション識別子にはインジケータが存在している
	0Ch		長さ
	RID		登録されたアプリケーションプロバイダ識別子 (RID) # A0 00 00 03 06
	SS		基準のバイト
	C0 ..C1h		カードネームバイト
	00 00 00 00h	RFU	RFU # 00 00 00 00
4+N	UU	TCK	T0からTkまでのすべてのバイトの排他的論理和



例：

MIFARE Classic 1KカードのATR = {3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6Ah}

その中：

長さ (YY) = 0Ch
RID = {{A0 00 00 03 06h} (PC/SCワークグループ)
基準 (SS) = 03h (ISO 14443A、3 パート)
カードネーム (C0 ..C1) = [00 01h] (MIFARE Classic 1K)
基準 (SS) = 03h : ISO 14443A、3 パート
= 11h : FeliCa

カードネーム (C0 ..C1)：

00 01 : MIFARE Classic 1K	00 38 : MIFARE Plus® SL2 2K
00 02 : MIFARE Classic 4K	00 39 : MIFARE Plus® SL2 4K
00 03 : MIFARE Ultralight®	00 30 : Topaz和Jewel
00 26 : MIFARE Mini®	00 3B : FeliCa
00 3A : MIFARE Ultralight® C	FF 28 : JCOP 30
00 36 : MIFARE Plus® SL1 2K	FF [SAK] : 定義されていないタグ
00 37 : MIFARE Plus® SL1 4K	

5.5.1.2. ATR フォーマット (ISO14443-4-3 PICC に適用)

バイト	数値	標記	説明			
0	3Bh	最初のヘッダー				
1	8Nh	T0	高いニブル8の意味は：TA1、TB1とTC1がなくて、TD1だけが続いている。 下位ニブル N は歴史的なバイトの数です（HistByte 0 - HistByte N-1）			
2	80h	TD1	高いニブル8の意味は：TA2、TB2とTC2がなくて、TD2だけが続いている。 下位ニブル 0 の意味は T=0			
3	01h	TD2	高いニブル0の意味は：TA3、TB3、TC3およびTD3が全部続いていない。 下位ニブル 1 の意味は T=1			
4 ~ 3+N	XX	T1	歴史バイト：			
	XX	Tk	ISO 14443-A： ATS応答の歴史的なバイト。ISO 14443-4基準を参照してください。			
			ISO 14443-B：			
			<table><tr><td>バイト 1~4</td><td>バイト 5~7</td><td>バイト8</td></tr><tr><td>ATQBのアプリケーションデータ</td><td>ATQBからのプロトコル情報バイト</td><td>高いニブル=ATTRIB コマンドのMBLI；下位ニブル（RFU）=0</td></tr></table>	バイト 1~4	バイト 5~7	バイト8
バイト 1~4	バイト 5~7	バイト8				
ATQBのアプリケーションデータ	ATQBからのプロトコル情報バイト	高いニブル=ATTRIB コマンドのMBLI；下位ニブル（RFU）=0				
4+N	UU	TCK	T0からTkまでのすべてのバイトの排他的論理和			

例1 :

MIFARE® DESFire®のATR = {3B 81 80 01 80 80h} // 6 バイトのATR

注釈 : APDU“FF CA 01 00 00h”を使用して、ISO 14443A-4のPICCに準拠しているまたはISO 14443B-4のPICCに準拠しているを区別します。可能な場合、完全なATSを取得します。ISO 14443A-3またはISO 14443B-3/4のPICCに準拠する場合、ATSが返される。

APDU コマンド = FF CA 01 00 00h

APDU 応答 = 06 75 77 81 02 80 90 00h

ATS = {06 75 77 81 02 80h}

例2 :

EZ-LinkのATR = {3B 88 80 01 1C 2D 94 11 F7 71 85 00 BEh}

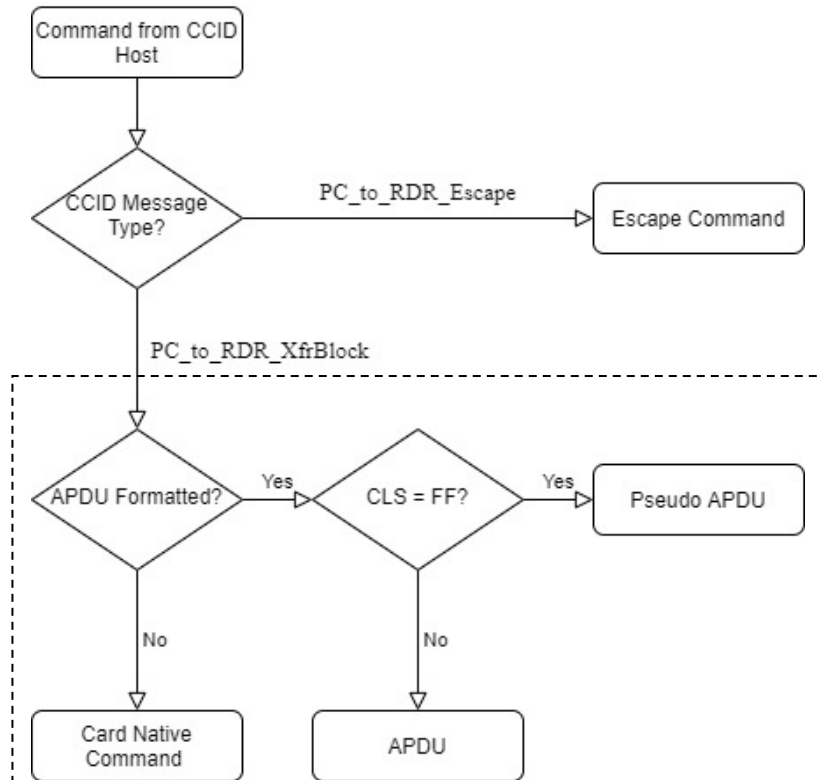
ATQBのアプリケーションデータ = 1C 2D 94 11h

ATQBのプロトコル情報 = F7 71 85h

ATTRIB の MBLI = 00h

5.5.2. APDU、私有 APDU およびカード固有コマンド

ユーザは、PC _ to _ RDR _ XfrBlock メッセージを介して、APDU、Private APDU（Pseudo APDU）、およびカード固有コマンド（Card Native Command）をリーダライタに送信することができます。コマンドの処理が完了すると、リーダライタは RDR _ to _ PC _ DataBlock メッセージを介してレスポンスを返します。



CCID ホストが PCSC API 中の SCardTransmit () に対応する CCID メッセージ PC _ to _ RDR _ XfrBlock を使用して、カードリーダーにカードアドミニストレーションコマンドまたは APDU を送信することができます。PICC では、カードが ISO 14443 第 4 部分プロトコルまたは Innovation プロトコルをサポートしている場合、リーダライタはプロトコルフレームにコマンド/APDU をパッケージ化してカードに直接送信し、コマンド/APDU を解析しません。カードが両方のプロトコルをサポートしていない場合、CCID ホストにメッセージ「6 A 81」が返されます。

注：Microsoft Window はスマートカードプラグアンドプレイに対応しているため、Microsoft Window はカードが提示された時にカードに APDU 命令を送信することがあります。この操作により、カードをリセットしない限り、DESFire カードが ISO APDU モードになり、固有コマンドを受信できなくなります。通常、Microsoft Window はカードが無反応状態になってから 10 秒後にカードをリセットします（PC _ to _ RDR _ lccPowerOff 経由）。

5.5.3. PICC の PCSC 私有 APDU（独自の拡張機能付き）

非接触カードへの間接アクセスには、以下の私有（Pseudo）APDU を使用します。CCID ホストが PCSC API 中の SCardTransmit () に対応する CCID メッセージ PC _ to _ RDR _ XfrBlock を使用して、カードリーダーにこれらの APDU を送信することができます。私有 APDU を受信すると、リーダライタは低レベルのコマンドを解読し、カードに送信します。カードがこれらの低レベルコマンドを処理した後、リーダライタはカードレスポンスを収集し、レスポンスを作成して CCID ホストに返信する。

5.5.3.1. データ取得（Get Data）[FFCA...]

このコマンドは、シーケンス番号、プロトコルパラメータなど、アクティブ化中に取得したデータを読み込むために使用します。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Data	FFh	CAh	下記の表を確認ください		00h (全長)

コマンドパラメーター

P1	P2	意味
00h	00h	カードの UID/PUPI/SN を取得する
01h	00h	A タイプ 4 パートの ATS を取得
02h	00h	下記のカードタイプ関連データを取得し、転送順序： A タイプ：2 バイト ATQA/ATVA + 4/7/10 バイト UID + 1 バイト最後の SAK。 B タイプ：12 バイト ATQB
80h	00h	下記のカードタイプ関連データを取得し、転送順序： A タイプ：2 バイト ATQA/ATVA + 4/7/10 バイト UID + 1/2/3 バイト SAK。 B タイプ：12 バイト ATQB FeliCa：17 バイト ATQ (+ 6 バイト ATTR、アクティブ化されている場合) SRI：8 バイト UID + 1 バイトチップ ID。 ISO15693：1 バイト DSFID + 8 バイト UID CTS：4 バイト SN + 2 バイト ATQT Innovatron：4 バイト SN + 1 バイトタブアドレス。

応答

応答	データ出力		
結果	データ	SW1	SW2

応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。

結果	SW1 SW2	意味
エラー	6X XXh	失敗

例：

“接続された PICC”のシリアルナンバーを取得します

UINT8 GET_UID[5] = {FF, CA, 00, 00, 00};

“接続された ISO 14443-A PICC”の ATS を取得します

UINT8 GET_ATS[5] = {FF, CA, 01, 00, 00};

5.5.3.2. キーロード (Load Key) [FF 82 ...]

このコマンドは、キーバッファ番号で指定された内部キーバッファにキーデータをロードするために使用します。鍵バッファは失いやすいストレージに属し、認証に使用されるコンテンツが含まれています。このコマンドはカードデータ転送をしません。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Load Authentication Keys	FFh	82h	00h	キーバッファ番号 (0-1)	キー長さ	キー

キー長さ/データ

カードタイプ	キー長さ(Lc)	キーデータ (転送/保存順番)
MIFARE Standard MIFARE Plus SL1	06h	6 バイト Crypto1 Key A/B。
MIFARE Plus SL1 MIFARE Plus SL2	16h	6 バイト Crypto1 Key A/B + 16 バイト AES Key。
MIFARE Plus SL2	06h	6 バイト暗号化 Crypto1 Key A/B。
MIFARE UltraLightC MIFARE DESFire	10h	16 バイト 2K3DES Key。

応答コード



結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	6X XXh	失敗

例：

// 失いやすいキーのメモリに 00h キーをロードする {FF FF FF FF FF FFh}。

APDU = {FF 82 00 00 06 FF FF FF FF FF FFh}

5.5.3.3. 認証 (Authenticate) [FF 86 00 00 05 ...]

このコマンドは、カードに認証プロセスを実行するために使用します。認証が成功すると、ユーザーは保護されたブロック/ページにアクセスできます。コマンドを送信する前に、ユーザーは Load Key コマンドを使用して正しいキーデータを指定したキーバッファ番号バッファにロードする必要があります。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Authenticate	FFh	86h	00h	00h	05h	下記の表を確認ください

コマンドデータ

バイト 0	バイト 1	バイト 2	バイト 3	バイト 4
01h	00h (RFU)	アドレス	キーのタイプ	キーバッファ番号

アドレスとキーのタイプ

カードタイプ	アドレス	キーのタイプ
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	00h~FFh : ブロック 0~255	60h : Crypto1 Key A 61h : Crypto1 Key B
MIFARE UltraLightC	00h (RFU)	80h : 2K3DES
MIFARE DESFire	00h~0Eh : DESFire キー番号 0~14	0Ah : 2K3DES

応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	6X XXh	失敗

セクター (16 個のセクター, 各セクターには 4 個の連続的なブロックが含まれている)	データブロック (3 つのブロック, 各には 16 バイト)	トレーラーブロック (1 つのブロック, 16 バイト)	
セクター0	00h – 02h	03h	} 1 KB
セクター1	04h – 06h	07h	
..	..	..	
..	..	..	
セクター14	38h – 0Ah	3Bh	
セクター15	3Ch – 3Eh	3Fh	

表12 : MIFARE Classic 1K カードのメモリマップ

セクター (32 個のセクター, 各セクターには 4 個の連続的なブロックが含まれている)	データブロック (3 つのブロック, 各には 16 バイト)	トレーラーブロック (1 つのブロック, 16 バイト)	
セクター0	00h ~ 02h	03h	} 1 KB
セクター1	04h ~ 06h	07h	
..			
..			
セクター30	78h ~ 7Ah	7Bh	
セクター31	7Ch ~ 7Eh	7Fh	

セクター (8 個のセクター, 各セクターには 16 個の連続的なブロックが含まれている)	データブロック (15 つのブロック, 各には 16 バイト)	トレーラーブロック (1 つのブロック, 16 バイト)	
セクター32	80h ~ 8Eh	8Fh	} 2 KB
セクター33	90h ~ 9Eh	9Fh	
..			
..			
セクター38	E0h ~ EEh	EFh	
セクター39	F0h ~ FEh	FFh	

表13 : MIFARE Classic 4K カードのメモリマップ

バイトナンバー	0	1	2	3	ページ
シリアルナンバー	SN0	SN1	SN2	BCC0	0
シリアルナンバー	SN3	SN4	SN5	SN6	1
内部/ロック	BCC1	Internal	Lock0	Lock1	2
OTP	OPT0	OPT1	OTP2	OTP3	3
データ読み取り/書き込み	Data0	Data1	Data2	Data3	4
データ読み取り/書き込み	Data4	Data5	Data6	Data7	5
データ読み取り/書き込み	Data8	Data9	Data10	Data11	6
データ読み取り/書き込み	Data12	Data13	Data14	Data15	7
データ読み取り/書き込み	Data16	Data17	Data18	Data19	8
データ読み取り/書き込み	Data20	Data21	Data22	Data23	9
データ読み取り/書き込み	Data24	Data25	Data26	Data27	10
データ読み取り/書き込み	Data28	Data29	Data30	Data31	11
データ読み取り/書き込み	Data32	Data33	Data34	Data35	12
データ読み取り/書き込み	Data36	Data37	Data38	Data39	13
データ読み取り/書き込み	Data40	Data41	Data42	Data43	14
データ読み取り/書き込み	Data44	Data45	Data46	Data47	15

512 ビット
または
64 バイト

表14 : MIFARE Ultralight カードのメモリマップ

例 :

// {TYPE A, キーナンバーの 00h}によって、ブロック 04h を認証します。PC/SC V2.01, 廃止されます

APDU = {FF 88 00 04 60 00h};

// {TYPE A, キーナンバーの 00h}によって、ブロック 04h を認証します。PC/SC V2.07

APDU = {FF 86 00 00 05 01 00 04 60 00h}

注 : MIFARE Ultralight のメモリは自由にアクセスできます。認証はいりません。

5.5.3.4. バイナリブロック読み取り（Read Binary Blocks） [FF B0...]

このコマンドは、指定されたブロック/ページアドレスの場所から指定されたバイトのデータを PICC から読み出すために使用します。カードの種類によっては、このコマンドを呼び出す前に、ユーザーはまず認証を行い、これらのブロック/ページへのアクセス権を取得する必要がある場合があります。

コマンド：

コマンド	CLA	INS	P1	P2	Le
Read Binary Blocks	FFh	B0h	モードとアドレス		更新していないバイト

P1/P2（モードとアドレス）

カードタイプ	P1[7:4] モード	P1[3:0] + P2[7:0] 開始アドレス（MSB が前）
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	00h : テールブロックをスキップ 08h : テールブロックを含む	000h~0FFh : ブロック 0~255
MIFARE UltraLight MIFARE UltraLightC	00h（保留）	000h~02Fh : ページ 0~47
SRIX4K/SRT512	00h（保留）	000h~07Fh : ブロック 0~127 0FFh : システムゾーン
PicoPass	00h（保留）	000h~0FFh : ブロック 0~255
ISO15693	00h（保留）	000h~0FFh : ブロック 0~255
Topaz/NFC Type-1 タブ	00h（保留）	000h~7FFh : バイトアドレス

Le（読み取り待ちのバイト数）

タイプ	バイト 0	バイト 1	バイト 2
短い	00h : 256 バイトを読み取る 01h~FFh : 1~255 バイトを読み取る	--	
長い	00h	0000h : 65536 バイトを読み取る 0001h~FFFFh : 1~65535 バイトを読み取る	

応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	6X XXh	失敗

例：

// バイナリブロック 04h から 16 バイトを読み取る (MIFARE Classic 1K または 4K)

APDU = FF B0 00 04 10h

// バイナリブロック 80h から 240 バイトを読み出す (MIFARE Classic 4K)

// ブロック 80 からブロック 8Eh まで (15 個ブロック)

APDU = FF B0 00 80 F0h

5.5.3.5. バイナリブロック更新 (Update Binary Blocks) [FF D6 ...]

このコマンドは、指定されたブロック/ページアドレスの場所から指定されたバイトのデータを PICC に書き込むために使用します。カードの種類によっては、このコマンドを呼び出す前に、ユーザーはまず認証を行い、これらのブロック/ページへのアクセス権を取得する必要がある場合があります。

カード内のブロック/ページへのデータの書き込みは、カードのセキュリティ設定 (例えば MIFARE カードの末尾ブロック) を変更する可能性があるため、特に注意してください。誤ったデータを書き込むか、操作が失敗すると、カードがロックされる可能性があります。カードロックされるリスクを軽減するために、セキュリティブロック/ページに関わる場合には、1 つの APDU コマンドを使用して複数のブロック/ページにデータを書き込むことはお勧めしません。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Update Binary Blocks	FFh	D6h	モードとアドレス		書き込み待ちバイト	データバイト

P1/P2 (モードとアドレス) と Write Size 一致 (ブロック/ページ容量)

カードタイプ	P1[7:4] モード	P1[3:0] + P2[7:0] 開始アドレス (MSB が前)	ブロック/ページの容量 (バイト)
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	0x0 : テールブロックをスキップ 0x8 : テールブロックを含む	000h~0FFh : ブロック 0~255	16
MIFARE UltraLight MIFARE UltraLightC	0x0 (保留)	000h~02Fh : ページ 0~47	4

カードタイプ	P1[7:4] モード	P1[3:0] + P2[7:0] 開始アドレス (MSB が前)	ブロック/ページの容量 (バイト)
SRIX4K/SRT512	0x0 (保留)	SRIX4K/SRT512	4
PicoPass	0x0 (保留)	PicoPass	8
ISO15693	0x0 (保留)	ISO15693	1~32
Topaz/NFC Type-1 タブ	0x0 : 消去を含む 0x0 : 消去を含まない	000h~7FFh : バイトアドレス	1(アドレス 78h)また 8(その他)

Lc (書き込み待ちバイト)

タイプ	バイト 0	バイト 1	バイト 2
短い	01h~FFh : 1~255 バイト書き込み待ち	--	
長い	00h	0001h~FFFFh : 1~65535 バイト書き込み待ち	

応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	6X XXh	失敗

例 :

// MIFARE Classic 1K/4K カード中のバイナリブロック 04h のデータを{00 01 ..0Fh}に更新します

APDU = {FF D6 00 04 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0Fh}

//MIFARE Ultralight 中のバイナリブロック 04 h を{00 01 02 03}に更新する

APDU = {FF D6 00 04 04 00 01 02 03h}

5.5.4. 透過コマンド (Pass Through Command)

5.5.4.1. ISO14443-3 タグをアクセスする

このセクションでは、ISO 14443-3 に準拠したコマンドを送信するために透過コマンド形式を使用する方法を示します。
ファームウェアの要件は次のとおりです：

- ACR1555U ファームウェアバージョン 1.3.1

コマンド

タイプ	Class	INS	P1	P2	Lc	Data In
ISO14443-3 Command	FFh	00h	00h	00h	Data Length	Data

Response

Response	Data Out		
結果	Data	SW1	SW2

応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	6X XXh	失敗

5.5.4.1.1. ISO14443-3 ネイティブコマンドを透過コマンドに変換する

本節では ISO 14443-3 ネイティブコマンドを透過コマンドに変換する方法を説明します。

サンプル：

- Mifare Ultralight AES バージョン取得 (Get Version) コマンド：
60
- 透過フォーマット：
FF 00 00 00 01 60

5.5.4.2. ISO15693 タグをアクセスする

このセクションでは、ISO 15693 に準拠したコマンドを送信するために透過コマンドフォーマットを使用する方法を説明します。ファームウェアの要件は次のとおりです：

コマンドのフォーマット

コマンドタイプ	Class	INS	P1	P2	Lc	Data In		Le
ISO15693 Command	FFh	FBh	00h	Flags	Data Length+1	Command Code	Data	--/00h

説明：

- **Flags:**
 - 00H (デフォルト)：flags を 22h に設定して、データに UID を含める必要はありません
 - その他の値：指定した値を flags として使用

応答

応答	Data Out		
結果	Data	SW1	SW2

応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	6X XXh	失敗

5.5.4.2.1. ISO15693 ネイティブコマンドを透過コマンドに変換する

本節では ISO15693 ネイティブコマンドを透過コマンドに変換する方法を説明します。透過コマンドは、26 kbps レートで実行されるコマンドのみをサポートしています。53 kbps 対応コマンドについては、**PCSC 2.0 パート 3 (バージョン 2.02 以上) の APDU コマンドを参照してください。**

サンプル 1

Stay Quiet (Mandatory Command)

22 02 XX XX XX XX XX XX XX XX

Where:

22 Flags

02 Stay quiet command code

XX XX XX XX XX XX XX XX 8 bytes UID

Pass Through Format

FF FB 00 22 09 02 XX XX XX XX XX XX XX XX

Or

FF FB 00 00 01 02

Example 2:

Read Multiple Blocks with not addressed (Optional Command)

02 23 00 01

Where:

02 Flags

23 Read multiple blocks command code

00 First block number

01 Read 2 blocks

Pass Through Format

FF FB 00 02 03 23 00 01

Example 3:

Get Random Number in Selected state (Custom Command for NXP SLIX2)

12 B2 04

Where:

12 Flags

B2 Get random number command code

04 NXP manufacturer code

Pass Through Format

FF FB 00 12 02 B2 04

5.5.4.2.2. サポートコマンドリスト

次の表に、ISO 15693-3 仕様のすべてのコマンドをリストしています。「N/A」と表示されたコマンドは、テストが行われていないことを示します。これらのコマンドに関する質問があれば、info@acs.com.hk まで連絡してください。ISO 15693 カードはオプションのコマンドをすべてサポートしていない場合がありますので、対応するカードのデータマニュアルを参照してコマンドサポート状況を確認してください。

Command Code	Type	Function	Support
01h	Mandatory	Inventory	Yes
02h	Mandatory	Stay Quiet	Yes
20h	Optional	Read Single Block	Yes
21h	Optional	Write Single Block	Yes
22h	Optional	Lock Block	Yes
23h	Optional	Read Multiple Blocks	Yes
24h	Optional	Write Multiple Blocks	Yes
25h	Optional	Select	Yes
26h	Optional	Reset to Ready	Yes
27h	Optional	Write AFI	Yes
28h	Optional	Lock AFI	Yes
29h	Optional	Write DSFID	Yes
2Ah	Optional	Lock DSFID	Yes
2Bh	Optional	Get System Information	Yes
2Ch	Optional	Get Multiple Block Security Status	Yes
2Dh	Optional	Fast Read Multiple Blocks	No
30h	Optional	Extended Read Single Block	Yes
31h	Optional	Extended Write Single Block	Yes
32h	Optional	Extended Lock Block	Yes
33h	Optional	Extended Read Multiple Blocks	Yes
34h	Optional	Extended Write Multiple Blocks	Yes
35h	Optional	Authenticate	Yes
36h	Optional	KeyUpdate	N/A
37h	Optional	AuthComm Crypto Format Indicator	N/A
38h	Optional	SecureComm Crypto Format Indicator	N/A
39h	Optional	Challenge	No
3Ah	Optional	ReadBuffer	Yes
3Bh	Optional	Extended Get System Information	Yes
3Ch	Optional	Extended Get Multiple Block Security Status	Yes
3Dh	Optional	Fast Extended Read Multiple Blocks	No
A0-DFh	Custom	IC Mfg Dependent	N/A
E0-FFh	Proprietary	IC Mfg Dependent	N/A

5.5.4.3. FeliCa タグのアクセス

FeliCa タグのアクセスコマンドは PCSC および MIFARE カードのアクセスコマンドと異なっています。このコマンドは FeliCa 基準に準拠して、ヘッダが追加されています。

FeliCa コマンドのフォーマット

コマンド	CLA	INS	P1	P2	Lc	データイン
FeliCa コマンド	FFh	00h	00h	00h	データの長さ	FeliCa コマンド (長さバイトで始まる)

FeliCa の応答データフォーマット (データ+2 バイト)

応答	データ出力
結果	応答データ

例のメモリブロックデータの読み取り

1. FeliCa を接続する。

ATR = 3B 8F 80 01 80 4F 0C A0 00 00 03 06 **11 00 3B** 00 00 00 00 42h

その中 : **11 00 3Bh** = FeliCa

2. FeliCa IDM の読み取り。

コマンド = FF CA 00 00 00h

応答 = [IDM (8 バイト)] 90 00h

例 : FeliCa IDM = 01 01 06 01 CB 09 57 03h

3. FeliCa コマンドアクセス。

例 : メモリブロックデータの「読み取り」

コマンド = FF 00 00 00 10 10 06 **01 01 06 01 CB 09 57 03** 01 09 01 01 80 00h

その中 :

Felica コマンド = 10 06 **01 01 06 01 CB 09 57 03** 01 09 01 01 80 00h

IDM = **01 01 06 01 CB 09 57 03h**

応答 = メモリブロックデータ

5.5.5. PCSC 2.0 第3部がサポートする APDU コマンド (V2.02 以降)

PCSC 2.0 の第3部で規定されているコマンドは、アプリケーションから非接触タグへのデータの透過転送、受信データの透過返却、プロトコルの切替えを目的とします。

5.5.5.1. PCSC 2.0 第3部コマンドフロー図

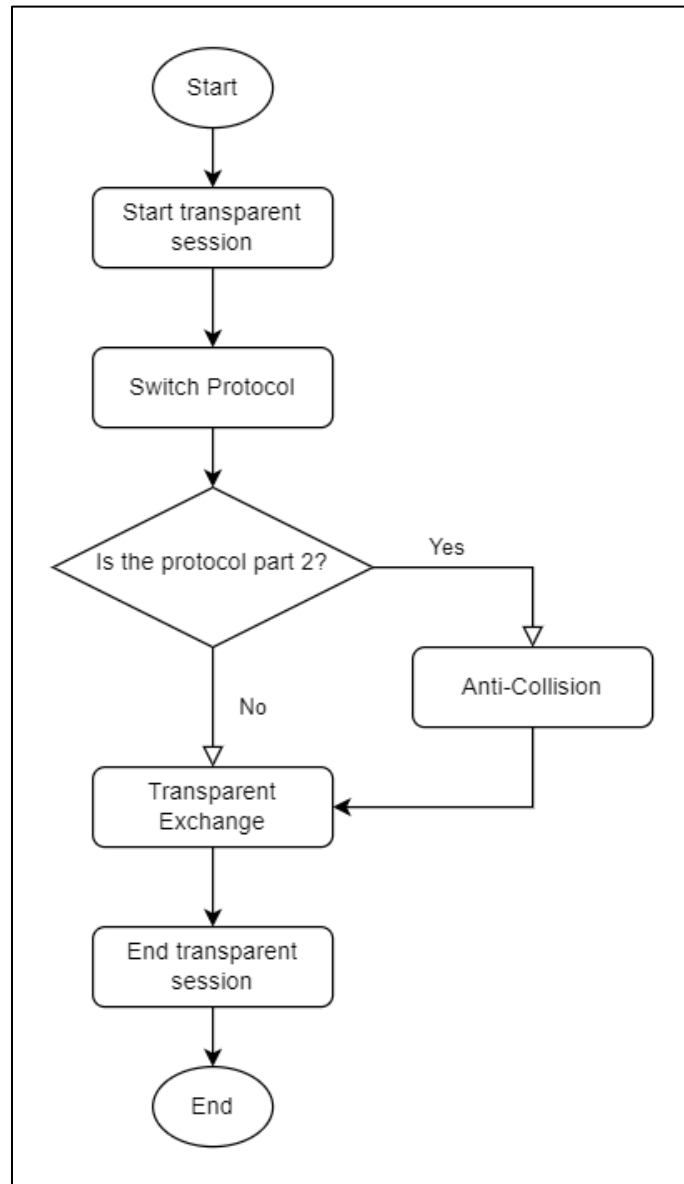


図8：透過セッションのフロー図

5.5.5.2. コマンドおよびレスポンスの APDU フォーマット

コマンドフォーマット

CLA	INS	P1	P2	Lc	コマンドデータフィールド
FFh	C2h	00h	機能	データ長	データ[データ長

機能 (1 バイト) :

- 00h = セッション管理
- 01h = 透過インタラクション
- 02h = プロトコル切替
- その他 = RFU (予約)

レスポンスフォーマット

レスポンスデータフィールド	SW1	SW2
BER-TLV で符号化されたデータ		

すべてのコマンドは、レスポンスデータフィールド (存在する場合) に加え、SW1 と SW2 を返します。SW1/SW2 は ISO 7816 に準拠し、以下の C0 データオブジェクトの SW1 SW2 も使用することができます。

C0 データ要素フォーマット

タグ	長さ (1 バイト)	SW2
C0h	03h	エラー状態

エラー状態説明

错误状态	说明
XX SW1 SW2	XX = APDU 内の不正なデータオブジェクトの番号 00 = APDU の一般的なエラー 01 = 第 1 データオブジェクトにエラーがある 02 = 第 2 データオブジェクトにエラーがある
00 90 00h	エラーなし
XX 62 82h	データオブジェクト XX の警告、情報が存在しない
XX 63 00h	情報なし
XX 63 01h	他データオブジェクト失敗のため中止
XX 6A 81h	データオブジェクト XX が未サポート
XX 67 00h	データオブジェクト XX の長さが不正
XX 6A 80h	データオブジェクト XX の値が不正
XX 64 00h	データオブジェクト XX 実行エラー (IFD 無応答)
XX 64 01h	データオブジェクト XX 実行エラー (ICC 無応答)
XX 6F 00h	データオブジェクト XX 失敗、詳細なし

最初のバイトはエラーのデータオブジェクト番号 XX、最後の 2 バイトがエラーの説明となります。ISO 7816 で規定された SW1 SW2 値も使用可能です。

APDU データフィールド内に複数のデータオブジェクトがある場合、1 つのデータオブジェクトが失敗しても、他が依存していなければ続けて処理可能です。

5.5.5.3. セッション管理 (Manage Session) [FF C2 00 00 ...]

このコマンドは、セッションを開始しポーリングを無効化することで、後続の通信を行えるようにします。通信完了後は、ユーザーが直ちにセッションを終了すべきです。

注意: このコマンドを誤用すると、リーダーがカードの有無を検知できなくなり、自動復旧せず物理的または論理的なリセットが必要となる場合があります

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータ	Le
Manage Session	FFh	C2h	00h	00h	Cmd 長	Cmd TLV	--/00h

レスポンスデータ

レスポンスデータ	SW1 SW2	意味
--	90 00h	操作成功
Rsp TLV	90 00h	Cmd TLV に失敗あり。詳細は Rsp TLV 参照
--	6X XXh	Cmd TLV のいずれかが失敗

Cmd TLV

Cmd	意味
Start Session: 81 00h	セッション開始、ポーリング無効化
RF Off: 83 00h	RF (アンテナ) オフ
Timer: 5F 46 04h [TIME]	次の RF On/Off TLV の前に設定する休止時間 [TIME] : 4 バイト値 (MSB が先頭)、範囲は 1000~100000 マイクロ秒 (us)。実際の休止時間は最も近い 1000us 単位に丸められます。
RF On: 84 00h	RF (アンテナ) オン
End Session: 82 00h	セッション終了、ポーリング再有効化

Rsp TLV

Rsp	意味
TLV Error: C0 03 NN 6X XXh	NN 番目のコマンド TLV でエラー

5.5.5.3.1. セッションデータオブジェクト (Start Session Data Object)

このコマンドは、透過セッション (トランスペアレントセッション) を開始するために使用します。セッションが開始されると、自動ポーリング機能はセッションが終了するまで無効化されます。

会話データオブジェクトの開始

タグ	長さ (1 バイト)	数値
81h	00h	-

5.5.5.3.2. セッションデータオブジェクト (End Session Data Object)

このコマンドはトランスペアレントセッションを終了するために使用します。新しいセッションが開始される前に、自動ポーリング状態にリセットされます。

会話データオブジェクトの終了

タグ	長さ (1 バイト)	数値
82h	00h	-

5.5.5.3.3. RF オフデータオブジェクト (Turn Off the RF Data Object)

このコマンドは、アンテナフィールドをオフにするために使用されます。

RF フィールドデータオブジェクトの閉鎖

タグ	長さ (1 バイト)	数値
83h	00h	-

5.5.5.3.4. RF オンデータオブジェクト (Turn On the RF Data Object)

このコマンドは、アンテナフィールドをオンにするために使用されます。

RF フィールドデータオブジェクトの開始

タグ	長さ (1 バイト)	数値
84h	00h	-

5.5.5.3.5. タイマーデータオブジェクト (Timer Data Object)

任意のビット/バイトをカードに送受信し、リンク層や伝送層（例：ISO14443-4）、冗長性（CRC 等）の設定も可能です。

注意: このコマンドはカードの内部状態を変更する場合があります、リセットやカードの再挿入が必要になることがあります。

タイマーデータオブジェクト



タグ	長さ (1 バイト)	数値
5F 46h	04h	计时器 (4 个字节)

5.5.5.4. 透過インタラクション (Transparent Exchange) [FF C2 00 01 ...]

任意のビット/バイトをカードに送受信し、リンク層や伝送層（例：ISO14443-4）、冗長性（CRC 等）の設定も可能です。

注意: このコマンドはカードの内部状態を変更する場合があります、リセットやカードの再挿入が必要になることがあります。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンド データ	Le
Transparent Exchange	FFh	C2h	00h	01h	Cmd データの長さ	Cmd TLV	00h

レスポンスステータスコード

結果	SW1 SW2	意味
成功	90 00h	成功。
エラー	6X XXh	失敗

Cmd TLV

Cmd	意味
Transceive Flag: 90 02 [Flag] 00h	次の Transceive TLV（送受信 TLV）の Flag を設定します。 Flag[7:5] : RFU（予約、将来使用のために予約）、0 に設定してください。 Flag[4] : ISO14443-4 を無効化する場合にセットします。 Flag[3] : 受信時のパリティ処理を禁止する場合にセットします。 Flag[2] : 送信時のパリティ処理を禁止する場合にセットします。 Flag[1] : 受信時の CRC 処理を禁止する場合にセットします。 Flag[0] : 送信時の CRC 処理を禁止する場合にセットします。 この TLV が省略された場合は、前回のコマンドで設定された Flag 値が使用されます。Flag 値が一度も設定されていない場合は、現在のプロトコルの値が使用されます。
Transmit Bit Frame: 91 01h [NumBit]	次の Transceive TLV の Bit Frame を設定します。この TLV が省略された場合、既定値は 0 です。 NumBit[7:3] : RFU（予約、将来使用）、0 に設定してください。 NumBit[2:0] : 最終バイトの有効ビット数を指定します（0 はすべてのビットが有効であることを示します）。
Timer: 5F 46 04h [TIME]	次の Transceive TLV のタイムアウト時間を設定します。 [TIME] : 4 バイト値（MSB が先頭）、範囲は 1us ~ 1,000,000us です。実際のタイムアウト時間は、最も近い $302.07 \times 2^0 \sim 2^{15} \text{ us}$ に丸められます。

Cmd	意味
	この TLV が省略された場合は、以前に設定された FWTI 値がタイムアウト時間として使用されます。
Set FWTI: FF 6E 03 03 01h [FWTI]	Transceive (送受信) の FWT/タイムアウトを設定します。 以前に「FF C2h ...」コマンドで FWTI が設定されていない場合、既定値は 0 となります。 - FWTI : 0~15 - FWT/タイムアウト : $302.07 \times 2^{\text{FWTI}}$ (マイクロ秒) (FWTI の値に応じて、FWT/タイムアウトが指数的に増加します)
Transceive: 95h [Size] [Data]	Size : BER-TLV 長データフィールド内でエンコードされたデータのサイズ Data : 送信するデータ

Rsp TLV

Rsp	意味
Receive Bit framing: 92 01h [NumBit]	NumBit[7:3] : RFU (予約、将来使用のため)、0 に設定してください。 NumBit[2:0] : 最終バイトの有効ビット数 (0 はすべてのビットが有効であることを示します)
Response: 97h [Size] [Data]	Size : BER-TLV 長さフィールドでエンコードされたデータのサイズ Data : 受信したデータ
Response Status: 96 02h [Status] 00h	Status[7:4] : RFU (予約、将来使用のため)。 Status[3] : フレームエラー。 Status[2] : パリティエラー。 Status[1] : RFU (予約、将来使用のため)。 Status[0] : CRC エラー。

5.5.5.4.1. 送信および受信フラグデータオブジェクト (Transmission and Reception Flag Data Object)

このコマンドは、以下の送信に対してフレーミングパラメータおよび RF パラメータを定義するために使用します。

送信および受信フラグデータオブジェクト

タグ	長さ (1 バイト)	数値		
		バイト 0		バイト 1
		ビット	説明	
90h	02h	0	0 – 送信データの後に CRC を追加する 1 – 送信データの後に CRC を追加しない	00h
		1	0 – 受信データに対して CRC チェックを行う 1 – 受信データに対して CRC チェックを行わない	

タグ	長さ (1 バイト)	数値		
		バイト 0		バイト 1
		ビット	説明	
		2	0 – 送信データにパリティビットを挿入する 1 – パリティビットを挿入しない	
		3	0 – 受信データにパリティビットが含まれていることを期待する 1 – 受信データにパリティビットが含まれていることを期待しない（つまり、パリティチェックを行わない）	
		4	0 – 送信データにプロトコルヘッダーを追加する、または応答から削除する 1 – プロトコルヘッダー（存在する場合）（例：PCB、CID、NAD）を追加または削除しない	
		5-7	RFU	

5.5.5.4.2. 送信ビットフレームデータオブジェクト (Transmission Bit Framing Data Object)

このコマンドは、送信または送受信予定のデータにおける最後のバイトの有効ビット数を定義するために使用します。

送信ビットフレームデータオブジェクト

タグ	長さ (1 バイト)	数値	
		ビット	説明
91h	01h	0-2	最後のバイトにおける有効ビット数（0 はすべてのビットが有効であることを示します）
		3-7	RFU

送信ビットフレームデータオブジェクトは、「送信」または「送受信」データオブジェクトと一緒にのみ使用できます。このデータオブジェクトが存在しない場合、すべてのビットが有効であることを示します。

5.5.5.4.3. 送受信データオブジェクト (Transceive Data Object)

このコマンドは、ICC（集積回路カード）との間でデータの送受信を行うために使用します。データの送信が完了すると、リーダー／ライターは、タイマーデータオブジェクトで指定された時間が経過するまで待機状態を維持します。

データフィールド内にタイマーデータオブジェクトが定義されていない場合は、設定済みの FWTI データオブジェクトで規定された時間だけ待機します。FWTI が設定されていない場合、リーダー／ライターは約 302μs 待機します。

送受信データオブジェクト

タグ	長さ (1 バイト)	数値
95h	データ長	データ (N バイト)

5.5.5.4.4. タイマーデータオブジェクト (Timer Data Object)

このコマンドは、1 マイクロ秒 (μ s) 単位の 32 ビットタイマーデータオブジェクトを作成するために使用します。

例えば、5000 μ s のタイマーデータオブジェクトが指定された場合、リーダー/ライターは後続の送受信 TLV の処理前に約 5000 μ s 待機し、その後でタイムアウトとなります。

タイマーデータオブジェクト

タグ	長さ (1 バイト)	数値
5F 46h	04h	計時器 (4 个字节)

5.5.5.4.5. レスポンスビット成枠データオブジェクト (Response Bit Framing Data Object)

このコマンドは、レスポンス内で受信した送信ビット成枠データオブジェクトを通知するために使用します。

タグ	長さ (1 バイト)	数値	
		ビット	説明
92h	01h	0-2	最后一个字节中的有效位数量 (0 表示所有的位都有效)
		3-7	RFU

送信ビットフレームデータオブジェクトは、「送信」または「送受信」データオブジェクトと一緒にのみ使用できます。このデータオブジェクトが存在しない場合、すべてのビットが有効であることを示します。

5.5.5.4.6. レスポンスステータスデータオブジェクト (Response Status Data Object)

このコマンドは、レスポンス内で受信したデータの状態を通知するために使用します。

応答ステータスデータオブジェクト

タグ	長さ (1 バイト)	数値		
		バイト 0		バイト 1
		ビット	説明	
96h	02h	0	0 – CRC が正しい、またはチェックを行っていない 1 – CRC チェックに失敗した	RFU

タグ	長さ (1 バイト)	数値		
		バイト 0		バイト 1
		ビット	説明	
		1	0 – コンフリクトなし 1 – コンフリクト検出	
		2	0 – パリティビットエラーなし 1 – パリティビットエラー検出	
		3	0 – フレーミングエラーなし 1 – フレーミングエラー検出	
		4 - 7	RFU	

5.5.5.4.7. レスポンスデータオブジェクト (Response Data Object)

このコマンドは、レスポンス内で受信したデータの状態を通知するために使用します。

応答データオブジェクト

タグ	長さ (1 バイト)	数値
97h	データ長	応答データ (N バイト)

5.5.5.5. プロトコル切 (Switch Protocol) [FF C2 00 02 ...]

このコマンドは、ユーザーがプロトコルの切り替えや指定、およびプロトコル層やパラメータの選択を行うことを可能にします。

注意点として、このコマンドはカードがサポートする内部処理に影響を与える場合があります。ドライバやファームウェアに通知することなくカードの状態を変更する可能性があります。そのため、ドライバやファームウェアを正常な状態に戻すためには、カードのリセットや取り外しが必要になることがあります。

命令

命令	CLA	INS	P1	P2	Lc	コマンド データフ ィールド	Le
Switch Protocol	FFh	C2h	00h	02h	Cmd 数据长度	Cmd TLV	00h

応答ステータスコード

応答データ	SW1 SW2	意味
Rsp TLV	90 00h	データ成功
--	90 00h	成功

応答データ	SW1 SW2	意味
--	6X XXh	失敗

Cmd TLV

Cmd	意味
Set Baud: FF 6E 03 05 01h [Baud]	プロトコル切り替え時に使用するパート 4 / レイヤ 4 のボーレートを設定します。[Baud]が「FF C2h ...」コマンドでまだ設定されていない場合、デフォルト値は 98h (106 kbps) です。 ISO14443: 98h (106 kbps), 99h (212 kbps), 9A (424 kbps), 9B (848 kbps). ISO15693: 80h (26 kbps), 08h (53 kbps)
Switch Protocol: 8F 02h [RF] [Layer]	プロトコルを指定された RF および / またはレイヤに切り替えます。 [RF]: 00h : ISO14443A, 01h : ISO14443B 02h : ISO15693, 03h : FeliCa, FFh : 現在の RF その他 : RFU [Layer]: 02h : 第 2 層 / パート, 03h : 第 3 層 / パート 04h : 第 4 層 / パート (A/B のみ使用可能) その他 : RFU 注: 第 2 層 / パートに切り替える場合は、必ずトランスペアレントセッション (ポーリング無効) 状態でなければなりません。

Rsp TLV

Rsp	意味
Response: 8Fh [Size] [Data]	Size: BER-TLV 長データフィールドにエンコードされたデータのサイズ Data: ATR (第 4 パートの場合)、最終 SAK (A タイプ第 3 パートの場合)、または ATQB 内の PI (B タイプ第 3 パートの場合)

5.5.5.5.1. プロトコル切替データオブジェクト (Switch Protocol Data Object)

このコマンドは、プロトコルおよびさまざまな標準層を指定するために使用します。

プロトコル切替データオブジェクト

タグ	長さ (1 字节)	数値	
		バイト 0	バイト 1
8Fh	02h	00h — ISO/IEC14443 タイプ A 01h — ISO/IEC14443 タイプ B 02h — ISO15693 03h — FeliCa その他 — RFU	02h — 第 2 層に切り替え 03h — 第 3 層に切り替えまたはアクティブ化 04h — 第 4 層をアクティブ化 その他 — RFU

5.5.5.5.2. レスポンスデータオブジェクト (Response Data Object)

このコマンドは、レスポンス内で受信したデータの状態を通知するために使用します。

応答ステータスコード

タグ	長さ (1 バイト)	数値
5F 51h	データ長	ATR
8Fh	データ長	最終 SAK (A タイプ第 3 パートの場合)、または ATQB 内の PI (B タイプ第 3 パートの場合)。

5.5.5.6. PCSC 2.0 第 3 部コマンド例

- 透過セッションの開始
コマンド : FF C2 00 00 02 81 00
レスポンス : C0 03 00 90 00 90 00
- アンテナフィールドのオフ
コマンド : FF C2 00 00 02 83 00
レスポンス : C0 03 00 90 00 90 00
- アンテナフィールドのオン
コマンド : FF C2 00 00 02 84 00
レスポンス : C0 03 00 90 00 90 00
- ISO 14443-4A のアクティベート
コマンド : FF C2 00 02 04 8F 02 00 04
レスポンス : C0 03 01 64 01 90 00 (カードが存在しない場合)
C0 03 00 90 00 5F 51 [Len] [ATR] 90 00
- PCB を 0Ah に設定し、データ転送で CRC、パリティ、プロトコルヘッダを有効化
コマンド : FF C2 00 01 0A 90 02 00 00 FF 6E 03 07 01 0A
レスポンス : C0 03 00 90 00 90 00
- APDU “80B2000008” をカードに送信し、レスポンスを取得
コマンド : FF C2 00 01 0E 5F 46 04 40 42 0F 00 95 05 80 B2 00 00 08
レスポンス : C0 03 00 90 00 92 01 00 96 02 00 00 97 0C [カード応答] 90 00
- 透過セッションの終了
コマンド : FF C2 00 00 02 82 00
レスポンス : C0 03 00 90 00 90 00

5.5.6. PICC の専属の私有 APDU

以下の私有 (Pseudo) APDU は、PCSC Pseudo APDU への追加として、非接触カードへの間接アクセスに使用されます。これらの APDU の内部処理プロセスは PCSC Pseudo APDU と同様です。

5.5.6.1. 値ブロック書き込み (Write Value Block) [FF D7 ...]

このコマンドは、MIFARE 規格に準拠したカードのブロックに 4 バイトの値を書き込むために使用します。このコマンドを呼び出す前に、ユーザーはまず認証を行い、このブロックへのアクセス権を取得する必要があります。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Write Value Block	FFh	D7h	00h	ブロック番号	05h	下記の表を確認ください



コマンドデータ

バイト 0	バイト 1	バイト 2	バイト 3	バイト 4
00h	4 バイト値 (MSB の前)			

例 1 : Decimal -4 = {FFh, FFh, FFh, FCh}

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

例 2 : Decimal 1 = {00h, 00h, 00h, 01h}

VB_Value			
MSB			LSB
00h	00h	00h	01h

応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	6X XXh	失敗

5.5.6.2. 値ブロック読み取り (Read Value Block) [FF B1 ...]

このコマンドは、MIFARE 規格に準拠したカードの有効ブロックから 4 バイトの値を読み取るために使用します。このコマンドを呼び出す前に、ユーザーはまず認証を行い、このブロックへのアクセス権を取得する必要があります。

コマンド

コマンド	CLA	INS	P1	P2	Le
Read Value Block	FFh	B1h	00h	ブロック番号	04h

例 1 : Decimal -4 = {FFh, FFh, FFh, FCh}

数値			
MSB			LSB
FFh	FFh	FFh	FCh

例 2 : Decimal 1 = {00h, 00h, 00h, 01h}

数値			
MSB			LSB
00h	00h	00h	01h

応答

応答データ	SW1 SW2	意味
4 バイト値 (MSB の前)	90 00h	データ成功。
--	6X XXh	失敗

5.5.6.3. 値の減少/増加 (Decrement/Increment Value) [FF D7 ...]

このコマンドは、ソースブロックから 4 バイトの値を減少/増加させ、結果をターゲットブロックに格納するために使用します (カードは MIFARE 規格に準拠している必要があります)。結果を同じソースブロックに格納する場合は、ターゲットブロックの番号を 0 またはソースブロック番号に設定できます。このコマンドを呼び出す前に、ユーザーはまず認証を行い、ソースブロックとターゲットブロックへのアクセス権を取得する必要があります。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Decrement/Increment Value	FFh	D7h	ターゲットブロック#	ソースブロック#	05h	下記の表を確認ください

コマンドデータ

バイト 0	バイト 1	バイト 2	バイト 3	バイト 4
01h	4 バイト追加値 (MSB の前)			
02h	4 バイト減少値 (MSB の前)			

応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	6X XXh	失敗

5.5.6.4. 値ブロックコピー (Copy Value Block) [FF D7 ...]

このコマンドは、ソースブロックから値をターゲットブロックにコピーするために使用します（カードは MIFARE 規格に準拠している必要があります）。このコマンドを呼び出す前に、ユーザーはまず認証を行い、ソースブロックとターゲットブロックへのアクセス権を取得する必要があります。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Copy Value Block	FFh	D7h	00h	ソースブロック #	02h	下記の表を確認ください

コマンドデータ

バイト 0	バイト 1
03h	ターゲットブロック #

応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	6X XXh	失敗

5.5.7. PCSC 準拠のタグをアクセスする (ISO 14443-4)

すべての ISO14443-4 に準拠したカード (PICC カード) は、ISO7816-4 の APDU を理解できます。ACR1555U カードリーダーは ISO 7816-4 基準の APDU および応答を交換することによって、ISO14443-4 基準のカードと通信することができます。ACR1555U が内部で ISO14443 の 1 – 4 パートのプロトコルを処理します。

MIFARE Classic (1K/4K)、MIFARE Mini および MIFARE Ultralight タグは T=CL エミュレーションを介してサポートされます。MIFARE タグを標準な ISO 14443-4 タグとして取り扱えばいいです。詳しい情報が **PICC の PCSC 私有 APDU (独自の拡張機能付き)** を参照してください。

ISO 7816-4 APDU フォーマット

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン	Le
ISO 7816 4 パートのコマンド					データの長さ		応答データの予想の長さ

ISO 7816-4 仕様の応答データフォーマット (データ+2 バイト)

応答	データ出力		
結果	応答データ	SW1	SW2

一般的な ISO 7816-4 コマンドの応答コード

結果	SW1 SW2	意味
成功	90 00h	操作が成功に完了しました。
エラー	63 00h	操作が失敗しました。

典型的なシーケンスは：

1. タグを提出して、PICC インターフェースと接続します。
2. タグ中の情報を読み取り/更新する。

これを実行します：

1. タグと接続する。

タグの ATR は 3B 88 80 01 00 00 00 00 33 81 81 00 3Ah です。

その中、

ATQB アプリケーションのデータ = 00 00 00 00、ATQB プロトコル 情報 = 33 81 81。これは ISO 14443-4 Type B タグです。

2. APDU を送信して、乱数を入手する。



00 84 00 00 08

>> 1A F7 F3 1B CD 2B A9 58h [90 00h]

注：ISO 14443-4 Type A のタグに対して、APDU“FF CA 01 00 00h”によって ATS を入手できます。

例：

// ISO 14443-4 Type B PICC (ST19XR08E) から 8 バイトを読み取ります。

APDU = {80 B2 80 00 08h}

CLA = 80h

INS = B2h

P1 = 80h

P2 = 00h

Lc = なし

データ = なし

Le = 08h

応答：00 01 02 03 04 05 06 07h [\$9000h]

5.5.8. MIFARE DESFire タグをアクセスする (ISO 14443-3)

MIFARE® DESFire®は ISO7816-4 APDU パッケージモードとネイティブモードをサポートできます。MIFARE® DESFire®タグが活性化されると、MIFARE® DESFire®タグコマンドに送られた最初の APDU が“モード”を決定します。例えば最初の APDU が“ネイティブモード”を選択したら、残りの APDU コマンド必ず“ネイティブモード”です。それと同じ、最初の APDU が“ISO 7816-4 APDU パッケージモード”を選択する場合、残りの APDU は必ず“ISO 7816-4 APDU パッケージモード”です。

例 1：MIFARE® DESFire® ISO 7816-4 APDU パッケージモード

//ISO 14443-4 Type A PICC (MIFARE® DESFire®)から 8 バイトの乱数を読み取ります

APDU = {90 0A 00 00 01 00 00}

CLA = 90h; INS = 0Ah (MIFARE DESFire 指令); P1 = 00h; P2 = 00h

Lc = 01h; データ入力 = 00h; Le = 00h (Le = 00h 最大の長さを示す)

応答：7B 18 92 9D 9A 25 05 21 [\$91AF]

#ステータスコード{91 AFH} は MIFARE® DESFire®仕様で定義されています。詳細については、MIFARE® DESFire®仕様を参照してください。



例 2 : MIFARE DESFire ページング・リンク (ISO 7816 APDU パッケージモード)

// この例では、アプリケーションは“フレームレベルの連鎖”と関わっています。

//MIFARE® DESFire®カードのバージョン番号を取得するために。

ステップ 1 : APDU {90 60 00 00 00}を送信して、最初のフレームを取得します。INS=60h

応答 : 04 01 01 00 02 18 05 91 AF [\$91AF]

ステップ 2 : APDU {90 AF 00 00 00}を送信して、二番目のフレームを取得します。INS=AFh

応答 : 04 01 01 00 06 18 05 91 AF [\$91AF]

ステップ 3 : APDU {90 AF 00 00 00}を送信して、最後のフレームを取得します。INS=AFh

応答 : 04 52 5A 19 B2 1B 80 8E 36 54 4D 40 26 04 91 00 [\$9100]

5.5.9. サポート PICC ATR

デフォルトでは、次の PICC タイプ/技術がサポートされています。リーダライタにカードをタッチすると、PC _ to _ RDR _ IccPowerOn コマンドは次の ATR を CCID ホストに返します。

カードタイプ/技術	ATR
MIFARE Std 1k6	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6A
MIFARE Std 4k6	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 02 00 00 00 00 69
MIFARE UltraLight6	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 03 00 00 00 00 68
MIFARE Plus SL1 2k6	デフォルト : MIFARE Std 1kと同じ 予備 : 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 36 00 00 00 00 5D
MIFARE Plus SL1 4k6	デフォルト : MIFARE Std 4kと同じ 予備 : 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 37 00 00 00 00 5C
MIFARE Plus SL2 2k	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 38 00 00 00 00 53
MIFARE Plus SL2 4k	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 39 00 00 00 00 52
MIFARE UltraLight C6	デフォルト : 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 3A 00 00 00 00 51 予備 : MIFARE UltraLightと同じ
SmartMX、模擬 MIFARE Std 1k6	デフォルト : MIFARE Std 1kと同じ 予備 : ISO14443 -4、Type A 同じ



カードタイプ/技術	ATR
SmartMX、模擬 MIFARE Std 4k ⁶	デフォルト : MIFARE Std 4kと同じ 予備 : ISO14443 -4、Type A 同じ
ISO14443 -4、Type A	3B 8n 80 01 T1 ..Tn Tck n = ATS 内履歴バイト数 T1 ..Tn = ATS 内履歴バイト数 Tck = xor 8n 80 01 T1 ..Tn
ISO14443 -4、Type B	3B 88 80 01 T1 ..T8 Tck T1 ..T4 = ATQB 中のアプリケーションデータ T5 ..T7 =ATQB 中のプロトコル情報 T8 = ATA 中の MBLI Tck = xor 88 80 01 T1 ..T8
FeliCa	3B 8F 80 01 80 4F 0C A0 00 00 03 06 11 00 3B 00 00 00 00 42
ISO15693-3 Generic	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 00 00 00 00 63
Infineon My-D Vicinity (SRF55Vxxx)	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 0E 00 00 00 00 6D
ST LRI	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 13 00 00 00 00 70
NXP I-Code SLI	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 14 00 00 00 00 77
NXP I-Code SLIX/SLIX2	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 35 00 00 00 00 56
PicoPass 2K	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 17 00 00 00 00 79
PicoPass 2KS	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 18 00 00 00 00 76
PicoPass 16K	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 19 00 00 00 00 77
PicoPass 16KS	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1A 00 00 00 00 74
PicoPass 16K (8 x 2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1B 00 00 00 00 75
PicoPass 16KS (8 x 2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1C 00 00 00 00 72
PicoPass 32KS (16 + 16)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1D 00 00 00 00 73
PicoPass 32KS (16 + 8x2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1E 00 00 00 00 70
PicoPass 32KS (8x2 + 16)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1F 00 00 00 00 71
PicoPass 32KS (8x2 + 8x2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 20 00 00 00 00 4E

ACS の Android ライブラリを使用

² ACS 定義された OS または iPadOS ライブラリを使用する

予備の ATR 定義のキャンセル。

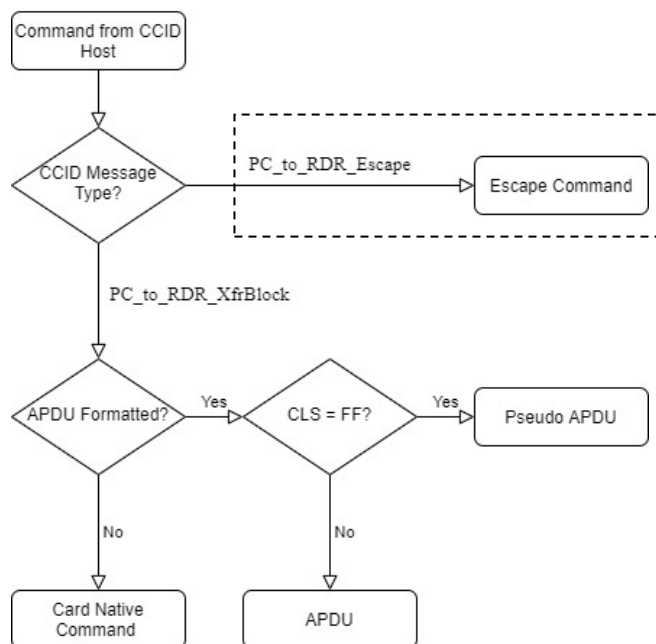


一般的なアプリケーションの応答時間を短縮するために、デフォルトでは次の PICC タイプ/テクノロジーのサポートが無効になっています。ユーザーは、直接コマンド“Set operation Mode”で、さまざまなタイプ/テクノロジーのサポートを有効にすることができます。対応するタイプ/テクノロジーが有効になり、リーダライタにカードをタッチ、PC _ to _ RDR _ lccPowerOn コマンドは次の ATR を CCID ホストに返します。

カードタイプ/技術	ATR
SRI (SRIX4K/SRT512)	3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 07 00 00 00 69
Topaz	3B 8F 80 01 80 4F 0C A0 00 00 03 06 02 00 30 00 00 00 5A
PicoPass 2K	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 17 00 00 00 75
PicoPass 2KS	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 18 00 00 00 7A
PicoPass 16K	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 19 00 00 00 7B
PicoPass 16KS	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1A 00 00 00 78
PicoPass 16K (8 x 2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1B 00 00 00 79
PicoPass 16KS (8 x 2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1C 00 00 00 7E
PicoPass 32KS (16 + 16)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1D 00 00 00 7F
PicoPass 32KS (16 + 8x2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1E 00 00 00 7C
PicoPass 32KS (8x2 + 16)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1F 00 00 00 7D
PicoPass 32KS (8x2 + 8x2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 20 00 00 00 42
Innovatron	3B 88 80 01 80 4F 05 F0 49 4E 4E 4F 35
CTS	3B 87 80 01 80 4F 04 F0 43 54 53 79

6.0. Escape コマンド

(Escape) コマンドは、PC_to_RDR_Escape (PCSC API の SCARD_CTL_CODE (3500) に対応する SCardControl()) を介して送信されます。コマンドの処理完了すると、リーダライタは RDR_to_PC_Escape メッセージを介してレスポンスを返します。



以下のコマンドは、PCD/NFC の構成、およびリーダライタへのアクセスのための特別な機能に使用されます。CCID ホストが PCSC API の SCARD_CTL_CODE (3500 (SCardControl)) に対応する CCID メッセージ PC_to_RDR_Escape を使用して、カードリーダーにこれらのコマンドを送信することができます。Escape コマンドを受信すると、リーダライタはコマンドを解釈して実行し、レスポンスを生成して CCID ホストに返信します。

注釈：

これらのコマンドは正しいインタフェースを介して送信する必要があります。例えば、E 0 00 25 01 00 (6.4.1節) は PICCインタフェースを介して送信されるべきです (6.0節)。

6.1. PICCEscape コマンド

6.1.1. RF 制御 (RF Control) [E0 00 00 25 01 ...]

このコマンドは RF 制御を設定するために使われます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
RF Control	E0h	00h	00h	25h	01h	RF 状態

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	RF 状態

RF 状態 : 1 バイト

RF 状態	説明
00h	RF オフ
01h	RF オン ポーリング
02h	RF オン ポーリングしない

デフォルト設定 – 01h (RF オン ポーリング)

6.1.2. PCD/PICC 状態取得 (Get PCD/PICC Status) [E0 00 00 25 00]

このコマンドは PCD/PICC の状態を取る際に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get PCD/PICC Status	E0h	00h	00h	25h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	Get PCD/PICC Status

PCD/PICC 状態 : 1 バイト

RF 状態	説明
00h	RF オフ
01h	PICC ない
02h	PICC 準備完了
03h	PICC 選択済み/アクティブ
FFh	エラー

6.1.3. ポーリング/ATR オプションの取得 (Get Polling/ATR Option) [E0 00 00 23 00]

このコマンドはポーリングオプションを設定/取得するために使用され、他のコマンドを使用することなく設定を保存できます。このコマンドは、最初のリーダ/ライタ構成にのみ使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Polling/ATR Option	E0h	00h	00h	23h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	03h	01h	PICC のポーリング/ATR オプション

6.1.4. ポーリング/ATR オプションの設定 (Set Polling/ATR Option) [E0 00 00 23 01 ...]

このコマンドはポーリングオプションを設置する時に使われます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Set Polling/ATR Option	E0h	00h	00h	23h	01h	PICC のポーリング/ATR オプション

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	PICC のポーリング/ATR オプション

PICC ポーリング/ATR オプション - 1 バイト

操作パラメータ	パラメーター	説明	オプション
Bit 0	ポーリングを有効する	PICCが検出待ちのタブタイプ をポーリングします。	1 = 検出
Bit 1	RF オフ間隔を有効にする		0 = スキップ
Bit 2	RFU		
Bit 3	追加の MIFARE タイプに対する第 3 部分カード ATR の識別を有効にする	PICC が検出待ちのタブタイプ をポーリングします。	1 = 検出
Bit 4 ~ 5	RF オフ		0 = スキップ
			下記の表を確認 ください
Bit 6	RFU		
Bit 7	SmartMX/JCOS カードシミュレーション MIFARE 用のパート 4 ATR を有効化	PICC が検出待ちのタブタイプ をポーリングします。	1 = 検出 0 = スキップ

RF オフ間隔-2 Bit ケース 1 : RF オフ無効 (Bit 1 = 0)

操作パラメーター		USB 実行(D0)
Bit 5	Bit 4	RF オフない
0	0	
0	1	
1	0	
1	1	

ケース 2 : RF オフ有効 (Bit 1 = 1)

操作パラメーター		USB 実行(D0)
Bit 5	Bit 4	250 ms 500 ms 1000 ms 2500 ms
0	0	
0	1	
1	0	
1	1	

デフォルト設定-8 Bh (ポーリングを有効にし、RF オフを有効にし、第 3 部分カードが ATR における追加の MIFARE タイプの識別を有効にし、Rf オフ間隔 [00]、SmartMX/JCOS カードシミュレーション MIFARE に適用する第 4 部分 ATR を有効にする)

6.1.5. PICC ポーリングタイプ取得 (Get PICCPolling Type) [E0 00 01 20 00]

このコマンドは許可されるテクノロジー/ポーリングタイプを取得するために使用され、他のコマンドを使用することなく設定を保存できます。最初のリーダライタ構成にのみ使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get PICC Polling Type	E0h	00h	01h	20h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	02h	PICC ポーリングタイプ

6.1.6. PICC ポーリングタイプ設定 (Set PICC Polling Type) [E0 00 01 20 02 ...]

このコマンドは PICC ポーリングタイプを設置する時に使われます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力	
						バイト 1	バイト 0
Set PICC Polling Type	E0h	00h	01h	20h	02h	PICC ポーリングタイプ	

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン	
						バイト 1	バイト 0
結果	E1h	00h	00h	00h	02h	PICC ポーリングタイプ	

PICC ポーリングタイプ-2 バイト、LSB の前で、ビットマスクは次の通りです

バイト	操作パラメーター	パラメーター	説明	オプション
バイト 1	Bit 0	ISO 14443A	PICCが検出待ちのタブタイプをポーリングします。 RFUビットは0に設定してください。	1 = 検出 0 = スキップ
	Bit 1	ISO 14443B		
	Bit 2	FeliCa		
	Bit 3	RFU		
	Bit 4	Topaz		
	Bit 5	Innovatron		
	Bit 6	SRI/SRIX		
	Bit 7	RFU		
バイト 0	Bit 0	Picopass (ISO14443B)		
	Bit 1	Picopass (ISO15693)		
	Bit 2	ISO15693		
	Bit 3	CTS		
	Bit 4-7	RFU		

デフォルト設定- バイト 1 : 07h (ISO14443 A タイプ、ISO14443 B タイプ、FeliCa)

バイト 0 : 05h (Picopass (ISO14443B), ISO15693)

例 :

コマンド : E0 00 01 20 02 07 05

応答 : E1 00 00 00 02 07 05

ポーリングタイプ: バイト 1 = 07h = 0000 0111b = ISO14443A, ISO14443B, FeliCa

バイト 0 = 05h = 0000 0101b = Picopass (ISO14443B), ISO15693

6.1.7. 自動 PPS 取得 (Get Auto PPS) [E0 00 00 24 00]

PICC が認識されるたびに、リーダーは最大接続速度によっては定義変更れた PCD および PICC との間の通信速度を変更しようとします。カードが提案された接続速度をサポートしていない場合、リーダーはより遅い速度でカードと接続しようとします。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Auto PPS	E0h	00h	00h	24h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン	
結果	E1h	00h	00h	00h	02h	最高速度	現在速度

6.1.8. 自動 PPS 設定 (Set Auto PPS) [E0 00 00 24 01...]

このコマンドは自動的な PPS のを設置するために使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Set Auto PPS	E0h	00h	00h	24h	01h	最高速度

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン	
結果	E1h	00h	00h	00h	02h	最高速度	現在速度

PPSの速率

速率	説明
00h	106 kbps; 自動PPSが設定されていないと同じです。
01h	212 kbps



速率	説明
02h	424 kbps
03h	848 kbps

デフォルト設定 – 02h(424 kbps)

注釈：

- 1、通常、アプリケーションが使用中のPICCの最大接続速度を知っている必要があります。環境にも達成可能な最大速度に影響します。リーダーは提案されている通信速度をよるして、PICCと話をします。PICCや環境が提案されている通信速度の要件を満たしていない場合、PICCはアクセスできなくなります。
- 2、高いレート設定がリーダライタの動作に影響を与える場合は、低いレート設定に切り替えます。

6.1.9. PICC タイプ取得 (Read PICC Type) [E0 00 00 35 00]

PICC タイプを読み取る時にこのコマンドを使用します。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get PICC Type	E0h	00h	00h	35h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン	
結果	E1h	00h	00h	00h	02h	タイプ	状態

タイプ : 1 バイト

タイプ	説明
CCh	PICC ない
04h	Topaz
10h	MIFARE
11h	FeliCa
20h	Type A, Part 4
23h	Type B, Part 4
25h	Innovatron
28h	SRIX
30h	PicoPass
FFh	その他

状態 : 1 バイト

状態	説明
00h	RF オフ
01h	PICC ない
02h	PICC 準備完了
03h	PICC 選択済み/アクティブ
FFh	エラー

6.1.10. PICC- HID キーボードの Escape コマンド

6.1.10.1. 出力フォーマット取得 (Get Output Format) [E0 00 00 90 00]

このコマンドは RTC の出力フォーマットを取得する時に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Output Format	E0h	00h	00h	90h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン	
結果	E1h	00h	00h	00h	02h	出力フォーマット	出力順番

6.1.10.2. 出力フォーマット設定 (Set Output Format) [E0 00 00 90 02...]

このコマンドは出力フォーマットを設定する時に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力	
Set Output Format	E0h	00h	00h	90h	02h	出力フォーマット	出力順番

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン	
結果	E1h	00h	00h	00h	02h	出力フォーマット	出力順番

出力フォーマット 1 バイト。

操作パラメーター	パラメーター	説明	オプション
Bit 7 ~ 4	文字の大文字と小文字	PICCが検出待ちのタブタイプをポーリングします。	1 = 検出 0 = スキップ
Bit 3 ~ 0	表示モード		

出力順番 1 バイト。

状態	説明
00h	デフォルト順番(UIDバイト0, UIDバイト1 ... UIDバイトN) 例 : aa cc bb dd (原始/実際のUID順番)
01h	順番反転(UIDバイトN, UIDバイトN-1 ... UIDバイト0) 例 : dd bb cc aa (UID順番反転)
02h	順番反転 ISO 14443 & Felica 例 : dd bb cc aa (ISO 14443 & Felica UID順番反転)
04h	順番反転 ISO 15693 例 : dd bb cc aa (ISO 15693 UID順番反転)

大文字と小文字 : 高 4 位(Bit 7 から Bit 4 まで)

状態(bit 7 から bit 4 まで)	説明(x bit を注目する必要がない)
1XXX	保留
00x0	小文字
00x1	大文字
000x	4バイトのUIDのみサポート
001x	4、7、8、10バイトのUIDサポート

表示モード : 低 4 位(Bit 3 から Bit 0)

状態(bit 7 から bit 4 まで)	説明(x bit を注目する必要がない)
0h	Hex
1h	Dec (バイトごと)
2h	Dec
3h	6H-6H
4h	8H-8H
5h	10H-10H
6h	14H-14H



状態(bit 7 から bit 4 まで)	説明(x bit を注目する必要がない)
7h	20H-20H
8h	6H-8D
9h	6H-10D
Ah	8H-10D
Bh	10H-14D
Ch	2H4H-8D
Dh	14H-17D

6.1.10.3. UID 開始、中間、終了ビット文字の取得 (Get Character at Start, Between, at End UID) [E0 00 00 91 00]

このコマンドは UID 開始、中間、終了のビットを取得する際に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Character of UID	E0h	00h	00h	91h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン		
結果	E1h	00h	00h	00h	03h	中間	終わり	スタート

6.1.10.4. UID 開始、中間、終了ビット文字の設定 (Set Character at Start, Between, at End UID) [E0 00 00 91 03...]

このコマンドは UID 開始、中間、終了のビットを設定する際に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力		
Set Character of UID	E0h	00h	00h	91h	03h	中間	終わり	スタート

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン		
結果	E1h	00h	00h	00h	03h	中間	終わり	スタート

中間：1 バイト（各 UID 間のキャラクター）

状態	説明
FFh	中間にはキャラクターがない
その他	汎用シリアルバス（USB）HID使用テーブルを参照

終わり 1 バイト（末尾の文字を出力）

状態	説明
FFh	中間にはキャラクターがない
その他	汎用シリアルバス（USB）HID使用テーブルを参照

終始 1 バイト（開始文字を出力）

状態	説明
FFh	中間にはキャラクターがない
その他	汎用シリアルバス（USB）HID使用テーブルを参照

注釈：

- AZERTYキーボードレイアウトは、零(0)とバックスペースキー“ではなく,” “,” “,” “-”を中間の文字としてサポートされます。

6.1.10.5. キーボードレイアウト言語の取得（Get Keyboard Layout Language）[E0 00 00 92 00]

このコマンドはキーボードレイアウト言語を取得する時に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Keyboard Layout Language	E0h	00h	00h	92h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	キーボードレイアウト言語

6.1.10.6. キーボードレイアウト言語の設定（Set Keyboard Layout Language）[E0 00 00 92 01 ...]

このコマンドはキーボードレイアウト言語を設定する時に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Set Keyboard Layout Language	E0h	00h	00h	92h	01h	キーボードレイアウト言語

応答コード



応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	キーボードレイアウト言語

キーボードレイアウト言語：1 バイト

状態	説明
00h	英語
01h	フランス語
02h	保留
03h	リトアニア語

6.1.10.7. ホストインターフェース取得 (Get Host Interface) [E0 00 00 93 00]

このコマンドはホストインターフェースを取得する時に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Host Interface	E0h	00h	00h	93h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	ホストインターフェース

6.1.10.8. ホストインターフェース設定 (Set Host Interface) [E0 00 00 93 01...]

このコマンドはホストインターフェースを設定する時に使用されます。

お知らせ： 設定を適用するために、リーダーは自動的に電源を切ります。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Set Host Interface	E0h	00h	00h	93h	01h	ホストインターフェース

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	ホストインターフェース

ホストインターフェース:バイト

状態	説明
00h	HIDキーボードのみ適応
01h	CCIDカードリーダーのみ適用

デフォルト設定—01h

6.1.11. PICC-カードシミュレーションの Escape コマンド

6.1.11.1. カードエミュレーションモードに入る (Enter Card Emulation Mode) [E0 0000 00 40 03 ...]

このコマンドは、MIFARE Ultralight カードや FeliCa カードをエミュレートするために、リーダーをカードエミュレーションモードに設定するために使用されます。

注： Lock バイトは、エミュレートされた MIFARE Ultralight カードでサポートされていません。UID は、ユーザが設定可能です。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力		
Enter Card Emulation Mode	E0h	00h	00h : 臨時 01h : 保存	40h	03h	NFC モード	00h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	03h	NFC モード

NFC 設備モード : 3 バイト

状態	説明
02h	NFCフォーラムタイプ2ラベルモード
03h	FeliCa
その他	カードリーダー/ライターモード

注： 異なるカードシミュレーションモードに切り替える前に、まずカードの読み取り/書き込みモードに入ってください。カードシミュレーションモードの初期完了後にレスポンスが表示されます。

バイトナンバー	0	1	2	3	USB でバイトアドレスへのアクセス
シリアルナンバー	SN0	SN1	SN2	SN3	Nil

バイトナンバー	0	1	2	3	USB でバイトアドレスへのアクセス
保留	保留	保留	保留	保留	Nil
内部/ロック	保留	内部	Lock0	Lock1	Nil
データ読み取り/書き込み	Data0	Data1	Data2	Data3	0-3
データ読み取り/書き込み	Data4	Data5	Data6	Data7	4-7
データ読み取り/書き込み	Data8	Data9	Data10	Data11	8-11
データ読み取り/書き込み	Data12	Data13	Data14	Data15	12-15
データ読み取り/書き込み	Data16	Data17	Data18	Data19	16-19
データ読み取り/書き込み	Data20	Data21	Data22	Data23	20-23
データ読み取り/書き込み	Data24	Data25	Data26	Data27	24-27
データ読み取り/書き込み	Data28	Data29	Data30	Data31	28-31
データ読み取り/書き込み	Data32	Data33	Data34	Data35	32-35
データ読み取り/書き込み	Data36	Data37	Data38	Data39	36-39
データ読み取り/書き込み	Data40	Data41	Data42	Data43	40-43
データ読み取り/書き込み	Data44	Data45	Data46	Data47	44-47
データ読み取り/書き込み	Data48	Data49	Data50	Data51	48-51
データ読み取り/書き込み	Data52	Data53	Data54	Data55	52-55
データ読み取り/書き込み		...			...
データ読み取り/書き込み	Data1984	Data1985	Data1986	Data1987	1984-1987

表15 : NFC フォーラムタイプ 2 ラベルのメモリマップ (2000 バイト)

メモリ	1 データブロック (16 バイト)	USB でバイトアドレスへのアクセス
データリーダー/ライター	Block 0	0-15
データ読み取り/書き込み	Block 1	16-31
データ読み取り/書き込み	Block 2	32-47
データ読み取り/書き込み	Block 3	48-63
データ読み取り/書き込み	Block 4	64-79
データ読み取り/書き込み	Block 5	80-95
データ読み取り/書き込み	Block 6	96-111
データ読み取り/書き込み	Block 7	112-127
データ読み取り/書き込み	Block 8	128-143
データ読み取り/書き込み	Block 9	144-159

表16 : FeliCa カードのメモリマップ (160 バイト)

その中 :

デフォルト : ブロック 0 データ:{10h, 01h, 01h, 00h, 09h, 00h, 00h, 00h, 00h, 00h, 01h, 00h, 00h, 00h, 00h, 1Ch}

デフォルトブロック 0 のデータ NFC タイプ 3 タグの属性情報ブロック

注釈 :

1. FeliCa カードエミュレーションのサポートは暗号化せずに読み取り/書き込み。
2. FeliCa カード識別番号 (IDm) はユーザに定義可能で、メーカーコードが (0388) に固定されています。

6.1.11.2. カードエミュレーションのデータを読み取る (Read Card Emulation Data) (NFC フォーラムタイプ 2 ラベル) [E0 00 00 60 04 ...]

このコマンドはカードエミュレーションカードのデータを読み取るために使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン			
Read Card Emulation Data	E0h	00h	00h	60h	04h	00h	NFC モード	終始オフセット	長さ

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	長さ	データ

終始オフセット： 1 バイト – 表 15 中 Data0 からのアドレス

長さ： 1 バイト – バイト数量

6.1.11.3. カードエミュレーションのデータ書き込む (Write Card Emulation Data) (NFC フォーラムタイプ 2 ラベル) [E0 00 00 60 ...]

このコマンドは、エミュレートされたカードへの書き込みに使用されています。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン				
Write Card Emulation Data	E0h	00h	00h	60h	長さ + 04h	01h	NFC モード	終始オフセット	長さ	データ

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン			
結果	E1h	00h	00h	00h	03h	長さ	90h	00h	

NFC 設備モード：1 バイト

状態	説明
02h	NFC フォーラムタイプ 2 ラベルモード
03h	FeliCa
その他	カードリーダー/ライターモード

終始オフセット： 1 バイト – 表 15 中 Data0 からのアドレス

長さ： 1 バイト – バイト数量

6.1.11.4. カードエミュレーションのデータを読み取る（Read Card Emulation Data） （NFC フォーラムタイプ 2 ラベル）（拡張）

このコマンドはカードエミュレーションカードのデータを読み取るために使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc		コマンドデータイン			
Read Card Emulation Data	E0h	00h	01h	60h	05h	00h	NFC モード	終始オフセ ット Bit [15:8]	終始オフセ ット Bit [7:0]	長さ

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン				
結果	E1h	00h	00h	00h	長さ	データ				

終始オフセット： 2 バイト – **表 15** 中 SN 0 からアドレスを読み取る

長さ： 1 バイト – 読み取り待ちのバイト

6.1.11.5. カードエミュレーションのデータ書き込む（Write Card Emulation Data） （NFC フォーラムタイプ 2 ラベル）（拡張）

このコマンドは、エミュレートされたカードへの書き込みに使用されています。

コマンド

コマンド	CLA	INS	P1	P2	Lc		コマンドデータイン				
Write Card Emulation Data	E0h	00h	01h	60h	長さ + 05h	01h	NFC モード	終始オフセ ット Bit [15:8]	終始オフセ ット Bit [7:0]	長さ	データ

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン				
結果	E1h	00h	00h	00h	03h	長さ	90h	00h		

NFC 設備モード： 1 バイト

状態	説明
02h	NFC フォーラムタイプ 2 ラベルモード
その他	カードリーダ/ライタモード

終始オフセット： 2 バイト – **表 15** 中 SN 0 からアドレスを書き込む

長さ： 1 バイト – 書き込み待ちのバイト

6.1.11.6. NFC フォーラムタイプ 2 ラベル模擬 ID を設定 (Set Card Emulation of NFC Forum Type 2 Tag ID) [E0 00 00 61 03 ...]

このコマンドは、エミュレートされた MIFARE Ultralight の UID を設定するために使用されています。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Set Card Emulation Lock Data	E0h	00h	00h	61h	03h	3 バイトの UID

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	02h	90h 00h

6.1.11.7. NFC カードエミュレーションロックデータロックを設定する (Set Card Emulation Lock Data in NFC) [E0 00 00 65 01...]

このコマンドは NFC 通信中、カードエミュレーションのデータをロックするために使用されます。データがロックされると、NFC で書き換えることができません。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Set Card Emulation Lock Data	E0h	00h	00h	65h	01h	ロック

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	ロック

ロック：1 バイトーNFC 通信で書き換えされないように、データを保護します。

操作パラメーター	パラメーター	説明	オプション
Bit 7 ~ 2	保留	保留	
Bit 1	Felica ロックを有効にする	データは NFC 通信で書き換えられません。USB 直接コマンドでデータを変更できます。	0：ロックを無効にする 1：ロックを有効にする
Bit 0	NFC フォーラムタイプ 2 タブを有効にする		

6.1.11.8. カードエミュレーションの FeliCa の IDM を設定する (Set Card Emulation FeliCa IDm) [E0 00 00 64 06 ...]

このコマンドはカードエミュレーションの FeliCa カード上で、6 バイトの FeliCa カードフラグを設定するために使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	コマンドデータイン
Set Card Emulation FeliCa IDm	E0h	00h	00h	64h	06h	IDm

応答コード

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	06h	IDm

その中：

IDm 6 バイト

6.1.11.9. カードエミュレーション状態取得 (Get Card Emulation Status) [E0 00 00 69 00]

このコマンドは NFC 通信中、カードエミュレーションのデータを取得するために使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc
Get Card Emulation Status	E0h	00h	00h	69h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	状態

状態：1 バイト

操作パラメータ	モード	説明
Bit 7 ~ 6	保留	保留
Bit 5	模擬カードがアクティブ化された	1 = アクティブ化される
Bit 4	模擬カードが外れた	1 = カードが外れた
Bit 3	模擬カードが全部読み取りました	1 = すべてのデータが読み取りました
Bit 2	模擬カードが読み取りました	1 = データが読み取りました
Bit 1	模擬カード書き込まれた	1 = データ書き込まれた



操作パラメータ —	モード	説明
Bit 0	模擬カードが検出された	1 = カード検出



6.1.11.10. シミュレーション NFC フォーラムタイプ 2 ラベルモードのコマンドサンプルセット

このコマンドセットは、ACR1555U の NFC フォーラムタイプ 2 ラベルモードをシミュレートし、ACS サイトをトリガします
<https://www.acs.com.hk>。ステップ：

1. 次のコマンドを使用してカードシミュレーションモードに入ります。
 - 送信カードエミュレーションモードに入る (Enter Card Emulation Mode)

E0 00 00 40 03 02 00 00
2. 下記のコマンドで NDEF データを含めています。データ書きます：
 - 送信カードエミュレーションのデータ書き込む (Write Card Emulation Data) (NFC フォーラムタイプ 2 ラベル)

E0 00 00 60 1A 01 02 00 16 E1 10 F4 00 03 0F D1 01 0B 55 02 61 63 73 2E 63 6F 6D 2E 68 6B FE

コマンドセットは、長い URL アドレスの例をトリガします

<https://www.example.com/this/is/a/very/long/url/that/keeps/going/on/and/on/with/even/more/segments/added/to/make/sure/it/exceeds/the/typical/length/limit/of/260/bytes/which/is/surprisingly/easy/to/do/if/you/keep/adding/more/and/more/segments/like/this/one/and/even/more>

ACR1555U で NFC フォーラムタイプ 2 ラベルモードをシミュレーションします。。ステップ：

1. 次のコマンドを使用してカードシミュレーションモードに入ります。
 - カードエミュレーションモードに入る (Enter Card Emulation Mode) を送信する

E0 00 00 40 03 02 00 00
2. 下記のコマンドで NDEF データを書きます：
 - カードエミュレーションのデータ書き込む (Write Card Emulation Data) (NFC フォーラムタイプ 2 ラベル) コマンドを送信する NDEF メッセージは 256 バイトを超えるため、2 つの部分に分けて NFC フォーラムタイプ 2 タグに送信する必要があります。

E0 00 00 60 AC 01 02 00 A8 E1 10 F4 00 03 FF 01 09 C1 01 00 00 01 02 55 02 65 78 61 6D 70 6C 65 2E 63 6F 6D 2F 74 68 69 73 2F 69 73 2F 61 2F 76 65 72 79 2F 6C 6F 6E 67 2F 75 72 6C 2F 74 68 61 74 2F 6B 65 65 70 73 2F 67 6F 69 6E 67 2F 6F 6E 2F 61 6E 64 2F 6F 6E 2F 77 69 74 68 2F 65 76 65 6E 2F 6D 6F 72 65 2F 73 65 67 6D 65 6E 74 73 2F 61 64 64 65 64 2F 74 6F 2F 6D 61 6B 65 2F 73 75 72 65 2F 69 74 2F 65 78 63 65 65 64 73 2F 74 68 65 2F 74 79 70 69 63 61 6C 2F 6C 65 6E 67 74 68 2F 6C 69 6D 69 74 2F 6F 66 2F 32 36 30 2F 62 79 74

E0 00 00 60 6E 01 02 A8 6A 65 73 2F 77 68 69 63 68 2F 69 73 2F 73 75 72 70 72 69 73 69 6E 67 6C 79 2F 65 61 73 79 2F 74 6F 2F 64 6F 2F 69 66 2F 79 6F 75 2F 6B 65 65 70 2F 61 64 64 69 6E 67 2F 6D 6F 72 65 2F 61 6E 64 2F 6D 6F 72 65 2F 73 65 67 6D 65 6E 74 73 2F 6C 69 6B 65 2F 74 68 69 73 2F 6F 6E 65 2F 61 6E 64 2F 65 76 65 6E 2F 6D 6F 72 65 FE

注釈：

NDEF (NFC データ相互フォーマット) に関する詳細な情報と規定を了解する必要がある場合は、NDEF 仕様を参照することをお勧めします。この仕様では、NDEF レコードの構造と使用について包全面的なガイドラインと詳細情報を提供し、これらの NDEF レコードは NFC データのインタラクションによく使用されています。NDEF 仕様は、ACR1555U デバイス環境における NDEF コマンドとデータの解釈と利用方法を深く理解するのに役立ちます。

6.2. ICCEscape コマンド

6.2.1. カード電源取得（Get Card Power Configuration）[E0 00 00 0B 00]

このコマンドは、ICC カードの電源構成を取得するために使用され、最初のリーダ/ライタ構成のみに使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Card Power Configuration	E0h	00h	00h	0Bh	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	カード電源配置

6.2.2. カード電源配置設定（Set Card Power Configuration）[E0 00 00 0B 01...]

このコマンドはと ICC カードの電源配置を設定する時に使われます。最初のリーダ/ライタ構成にのみ使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Set Card Power Configuration	E0h	00h	00h	0Bh	01h	配置

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	カード電源配置

カードの電源配置（1 バイト）

カード電源配置	説明
00h	自動検出、1.8V -> 3V -> 5V
01h	5 V のみ
02h	3 V のみ
03h	1.8 V のみ
04h	自動検出、5V -> 3V -> 1.8V



カード電源配置	説明
その他	RFU

デフォルト設定 – 04h (自動検出、5V -> 3V -> 1.8V)

6.3. 周辺設備制御と他の Escape コマンド

6.3.1. ファームウェアバージョン取得（Get Firmware Version）[E0 00 00 18 00]

このコマンドはファームウェアのバージョンを入手する時に使われます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Firmware Version	E0h	00h	00h	18h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	ファームウェアのバージョンの長さ	ファームウェアのバージョン

例：

コマンド： E0 00 00 18 00

応答コード： E1 00 00 00 12 41 43 52 31 35 35 35 20 46 57 20 31 2E 30 30 2E 30 30

16進法ファームウェアバージョン： 41 43 52 31 35 35 35 20 46 57 20 31 2E 30 30 2E 30 30

ASCII ファームウェアのバージョン： ACR1555 FW 1.00.00

6.3.2. シリアルナンバー取得（Get Serial Number）[E0 00 00 47 00]

シリアルナンバーを取得する時にこのコマンドを使用します。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get Serial Number	E0h	00h	00h	47h	00h

応答コード

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	シリアルナンバーの長さ	シリアルナンバー

6.3.3. USB 記述子内の S/N を設定する (Set S/N in USB Descriptor) [E0 00 00 F0]

このコマンドは USB 記述子内の S/N を設定するために使われます。

コマンド

コマンド	CLA	INS	P1	P2	Le	コマンドデータイン	
Set S/N in USB Descriptor	E0h	00h	00h	F0h	02h	00h	USB 記述子内の SN を有効する

応答コード

応答	CLA	INS	P1	P2	Le	データ出力		
結果	E1h	00h	00h	00h	03h	USB 記述子内の SN を有効する	90h	00h

USB 記述子内の SN を有効する (1 バイト)

USB 記述子内の SN を有効する	説明
00h	USB 記述子内の SN を無効する
01h	USB 記述子内の SN を有効する

6.3.4. ブザー制御の設定-単発 (Set Buzzer Control - Single Time) [E0 00 00 28 01 ...]

このコマンドは単発のブザーを設定するために使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Buzzer Control	E0h	00h	00h	28h	01h	ブザー状態

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	ブザー状態

ブザーの状態(1 バイト)



ブザー状態	説明
00h	OFF
01 ~ FFh	オン、持続時間は 10 ms を単位にする

6.3.5. ブザー制御の設定-重複 (Set Buzzer Control - Repeatable) [E0 00 00 28 03 ...]

このコマンドはブザーの周期を設定するために使用されます

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Buzzer Control	E0h	00h	00h	28h	03h	ブザー状態

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	03h	ブザー状態

ブザーの状態(3 バイト)

操作パラメーター	ブザー状態	説明
パラメーター 1 - バイト 0	オン時間帯	01 ~ FF: オンにする持続時間は 10 ms を単位にする
パラメーター 2 - バイト 1	OFF 時間帯	01 ~ FF: OFF にする持続時間は 10 ms を単位にする
パラメーター 3 - バイト 2	繰り返し回数	01 ~ FF: 繰り返し回数

6.3.6. LED 状態取得 (Get LED Status) [E0 00 00 29 00]

このコマンドは LED の状態を取得するために使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get LED Status	E0h	00h	00h	29h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	LED 状態

6.3.7. LED 制御設定 (Set LED Control) [E0 00 00 29 01 ...]

このコマンドは LED 制御を設定するために使われます

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Set LED Control	E0h	00h	00h	29h	01h	LED 状態

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	LED 状態

LED 状態 (1 バイト)

LED 状態	説明
Bit 0 : 緑の LED	1 = ON ; 0 = OFF
Bit 2 = 青色 LED	1 = ON ; 0 = OFF
Bit 3 : 黄色の LED	1 = ON ; 0 = OFF
Bit 4-7 : RFU	その他

6.3.8. UI 操作取得 (Get UI Behaviour) [E0 00 00 21 00]

このコマンド PCD UI の操作を取得するために使用され、他のコマンドを使用することなく設定を保存できます。最初のリーダ/ライタ構成にのみ使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Le
Get PICC UI Behaviour	E0h	00h	00h	21h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	PICC UI 操作

6.3.9. UI 操作設定 (Set UI Behaviour) [E0 00 00 21 01 ...]

このコマンドは PICC UI 操作のパラメータを設定するために使われます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Set PICC UI Behaviour	E0h	00h	00h	21h	01h	PICC UI 操作

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	PICC UI 操作

UI 操作-1 バイト、ビットマスクは次の通りです

操作パラメーター	パラメーター	説明	オプション
Bit 0	動作中 (LED が速く点滅する)	リーダーの UI 操作	1 = 有効にする 0 = 無効にする
Bit 1	カードの提示待ち (LED 点灯)		
Bit 2	カードが存在/アクティブ化 (LED 点灯)		
Bit 3	カードがアンテナ領域に入るイベント (ブザーが一時的に鳴る)		
Bit 4	カードがアンテナ領域から外すイベント (ブザーが一時的に鳴る)		

PICC デフォルト設定 – 1Fh

注釈:

- UI 操作コマンドの取得/設定に SAM インタフェースは蓋まれていません。

6.3.10. BLE UI 操作取得 (Get BLE UI Behaviour) [E0 00 00 4B 01 05]

このコマンドは LED の操作を読み取る際に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Get BLE UI Behaviours	E0h	00h	00h	4Bh	01h	05h



応答コード

応答	CLA	INS	P1	P2	Le	データ	コマンドデータイン
結果	E1h	00h	00h	00h	02h	05h	BLEと充電 UI 操作

6.3.11. BLE UI 操作設定 (Set BLE UI Behaviour) [E0 00 00 4B 02 05 ...]

このコマンドは青色の BluetoothLED 操作を設定するために使われます。

コマンド

コマンド	CLA	INS	P1	P2	Lc	応答データ (1 バイト目)	データ出力
Set BLE UI Behaviour	E0h	00h	00h	4Bh	02h	05h	BLE UI 操作

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	BLE UI 操作

BLE UI 操作-1 バイト、ビットマスクは次の通りです

操作パラメーター	パラメーター	説明	オプション
Bit 0	青色 BLE LED	LEDがリーダーライターに制御される	1 = 有効にする 0 = 無効にする

BLE のデフォルト設定 - 01h

6.3.12. スリープモードオプションの取得 (Get Sleep Mode Option) [E0 00 00 50 00]

このコマンドはデバイスがスリープモードに入るタイマーを確認する時に使用されます。

注意：ファームウェアバージョン 1.02.04 以降でのみ使用可能

コマンド

コマンド	CLA	INS	P1	P2	Lc
Get Sleep Timer Option	E0h	00h	00h	50h	00h

応答コード



応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	時間

その中：

時間 1 バイト：タイマー

00h = 60 s（デフォルト）

01h = 90s

02h = 120s

03h = 180s

04h = スリープしない

6.3.13. スリープモードオプションの設定 (Set Sleep Mode Option) [E0 00 00 48 ...]

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入ります。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。

注意：ファームウェアバージョン 1.02.04 以降でのみ使用可能

コマンド

コマンド	CLA	INS	P1	P2	データ出力
Set Sleep Timer Option	E0h	00h	00h	48h	時間

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	時間

その中：

時間 1 バイト。時間
 00h = 60 s (デフォルト)
 01h = 90s
 02h = 120s
 03h = 180s
 04h = スリープしない

6.3.14. 自動シャットダウンオプション取得 [E0 00 00 0F 00]

このコマンドは、オートシャットダウンタイマーの設定を確認するために使用します。

注：この機能は、シリアル番号が RR662-XXXXXX 以上の機器でサポートされています。

コマンド

コマンド	Class	INS	P1	P2	Lc
Get Auto Power-Off Option	E0h	00h	00h	0Fh	00h

応答コード

応答	CLA	INS	P1	P2	Le	Data In
結果	E1h	00h	00h	00h	01 h	Time

6.3.15. 自動シャットダウンオプション設定 [E0 00 00 0F 01...]

自動シャットダウンされる前のタイム間隔を設定する時にこのコマンドを使用します。

注：この機能は、シリアル番号が RR662-XXXXXX 以上の機器でサポートされています。

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set Auto Power Off Option	E0h	00h	00h	0Fh	01 h	Time

Response Code

Response	CLA	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01 h	Time

Where:

Time 1 バイト: タイマー

00h = 禁止

01h = 5 分 (デフォルト)

02h = 10 分

03h = 15 分

6.3.16. 即時シャットダウン [E0 00 00 0E 00]

このコマンドにより、ACR1555U は直ちにシャットダウンされます。

注：この機能は、シリアル番号が RR662-XXXXXX (FW1.03.02) 以上の機器でサポートされています。

Command

Command	Class	INS	P1	P2	Lc
Set Auto Power Off Option	E0h	00h	00h	0Eh	00 h

Response Code

Response	CLA	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	02 h	90 00

6.3.17. Tx パワー数値取得 (Get Tx Power Value) [E0 00 00 51 00]

このコマンドは Bluetooth の Tx パワーを読み取る際に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc
Get Tx power Value	E0h	00h	00h	51h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	Tx パワー

6.3.18. Tx パワー数値設定 (Set Tx Power Value) [E0 00 00 49...]

このコマンドは Bluetooth の Tx パワーを修正する際に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	データ出力
Get Tx power Value	E0h	00h	00h	49h	Tx パワー

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	Tx パワー

その中：

Tx パワー 1 バイト

00h = -23 dBm、距離：～3 メーター

01h = -6 dBm (デフォルト)、距離：～7 メーター

02h = 0 dBm、距離：～17 メーター

03h = 4 dBm、距離：～25 メーター

デフォルト値 -01h

6.3.19. MAC アドレス取得 (Get MAC Address) [E0 00 00 43 00]

このコマンドは BLE リーダーの MAC アドレスを読み取る際に使用されます。

コマンド

コマンド	CLA	INS	P1	P2	Lc
Get MAC Address	E0h	00h	00h	43h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	06h	MAC アドレス

その中：

MAC アドレス 6 バイト

AA:BB:CC:DD:EE:FF (リトルエンドフォーマット)

BLE MAC アドレス : FF:EE:DD:CC:BB:AA

6.3.20. BLE 放送名称取得する (Get BLE Advertising Name) [E0 00 00 44 00]

BLE 放送名称を読み取る時にこのコマンドを使用します。

コマンド

コマンド	CLA	INS	P1	P2	Lc
Get BLE Advertising Name	E0h	00h	00h	44h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	length	BLE 放送名称

6.3.21. バッテリ取得 (Get Battery Level) [E0 00 00 52 00]

このコマンドは現在の電池残量検査するために使用されます。

注：Read Writer が Bluetooth モードになっている場合にのみ使用できます。

コマンド

コマンド	CLA	INS	P1	P2	Lc
Get Battery Level	E0h	00h	00h	52h	00h

応答コード

応答	CLA	INS	P1	P2	Le	コマンドデータイン
結果	E1h	00h	00h	00h	01h	電池残量

その中：

電池残量 電池残量 1 バイト

64h = 100%残量	32h = 50%残量
5Ah = 90%残量	28h = 40%残量
50h = 80%残量	1Eh = 30%残量
46h = 70%残量	14h = 20%残量
3Ch = 60%残量	0Fh = 15%残量

6.3.22. BLE バインドレコードの削除 (Remove BLE Bonding Record) [E0 00 00 5B 00]

BLE バインドレコードを削除するために使われます。注：USB モードのリーダーライターのみに適用します。

コマンド

コマンド	CLA	INS	P1	P2	Lc
Remove BLE Bonding Record	E0h	00h	00h	5Bh	00h

応答コード

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	02h	90h 00h

6.3.23. BLE 通信モード読み取る (Read BLE Communication Mode)

BLE 通信モードを読み取る時にこのコマンドを使用します。

コマンド

コマンド	CLA	INS	P1	P2	Lc
Read BLE Communication Mode	E0h	00h	00h	77h	00h

応答コード

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	01h	通信モード

6.3.24. BLE 通信モード設定 (Set BLE Communication Mode)

BLE 通信モードを設定する時にこのコマンドを使用します。

ステップ (1) 乱数を取得します。

コマンド

コマンド	CLA	INS	P1	P2	Lc
Get Random Number	E0h	00h	00h	75h	00h

応答コード

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	10h	16 バイトの乱数

ステップ (2) BLE 通信モードを設定する：

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Set BLE Communication Mode	E0h	00h	00h	77h	20h	暗号化された 16 バイト乱数 + 暗号化された 16 バイトモード値

その中：

モード値	
通信モード	15 バイト任意値

応答コード

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	01h	通信モード



通信モード	意味
00h	明文
01h	認証

6.3.25. 顧客マスターキーリライトー (Customer Master Key Rewrite)

顧客マスターキーを設置する時にこのコマンドを使用します。

ステップ (1) 乱数を取得します。

コマンド

コマンド	CLA	INS	P1	P2	Lc
Get Random Number	E0h	00h	00h	75h	00h

応答コード

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	10h	16 バイトの乱数

ステップ (2) 顧客マスタキーの上書き :

コマンド

コマンド	CLA	INS	P1	P2	Lc	データ出力
Customer Master Key Rewrite	E0h	00h	00h	76h	20h	暗号化された 16 バイト乱数 + 暗号された 16 バイトの新しいマスターキー

応答状態コード (成功)

応答	CLA	INS	P1	P2	Le	データ出力
結果	E1h	00h	00h	00h	02h	SW1 SW2

SW1 SW2	結果
90 00h	成功
67 00h	失敗

付録A. NDEF メッセージ

このセクションでは、NDEF メッセージを使用して NTag に URL をエンコードする方法について説明します。

このコマンドのデータフォーマットを了解したい場合、“NFC Forum NFC Data Exchange Format (NDEF) Specifications 1.0”を参照してください。

例：

NDEF メッセージ = { D1 01 0B 55 02 61 63 73 2E 63 6F 6D 2E 68 6Bh }

オフセット	コンテンツ	長さ	説明
0	D1	1	NDEF ヘッダ。TNF = 01h、SR=1、MB=1、ME=1
1	01	1	レコード名の長さ（1 バイト）
2	0B	1	URI ペイロードの長さ（11 バイト）
3	55 (“U”)	1	レコードタイプ“U”
4	02	1	略語“http://www.”
5	61 63 73 2E 63 6F 6D 2E 68 6B	10	URL 自体。“acs.com.hk”

Ntag にエンコード = { 03 0F D1 01 0B 55 01 61 63 73 2E 63 6F 6D 2E 68 6B FEh }

オフセット	コンテンツ	長さ	説明
0	03	1	TLV ヘッダ。03h = NDEF メッセージ
1	0F	1	NDEF メッセージの長さ（15 バイト）
2	D1 01 0B 55 01 61 63 73 2E 63 6F 6D 2E 68 6B	15	NDEF メッセージ
17	FE	1	TLV ヘッダ。FEh = 終了記録

付録B. ロットビット状態とスロットビットエラー

各 Bulk-IN メッセージには、スロットビットエラーとスロットビット状態レジスタの値が含まれています。

オフセット	データフィールド	大きさ	数値	説明
0	bmICCStatus	2 ビット	0、1、2	0 - ICC が存在し、アクティブ状態（安定給電がオン、RST 信号が非アクティブ） 1 - ICC は存在しますが非アクティブ（ハードウェアエラーのため非アクティブまたはオフ） 2 - ICC が存在しない 3 - RFU
2	bmRFU	4 ビット	RFU	スマートポスターデータの長さ（15 バイト）
6	bmCommandStatus	2 ビット	0、1、2	0 - 処理にエラーがない 1 - 失敗（エラーレジスタによるエラーコード） 2 - 時間延長のリクエスト 3 - RFU

表17：スロットビットステータスレジスタ

エラーコード	エラー名称	可能な原因
FFh	CMD_ABORTED	ホストが現在のアクティビティを中止しました
FEh	ICC_MUTE	ICC と通信する時、CCID がタイムアウト
FDh	XFR_PARITY_ERROR	ICC と通信する時にパリティエラー
FCh	XFR_OVERRUN	ICC と通信する時にオーバーランエラー
FBh	HW_ERROR	総合的なハードウェアエラーが発生しました
F8h	BAD_ATR_TS	
F7h	BAD_ATR_TCK	
F6h	ICC_PROTOCOL_NOT_SUPPORTED	
F5h	ICC_CLASS_NOT_SUPPORTED	
F4h	PROCEDURE_BYTE_CONFLICT	
F3h	DEACTIVATED_PROTOCOL	
F2h	BUSY_WITH_AUTO_SEQUENCE	自動シーケンス進行中

エラーコード	エラー名称	可能な原因
E0h	CMD_SLOT_BUSY	コマンドを処理しているスロットに 2 番目のコマンドを送信する
C0h から 81h まで	ユーザー定義	
80h 及び 必要要件 を埋める値	RFU	
7Fh から 01h まで	サポートされていないエラーメッセージパラメータのインデックス	01h : dwLength エラー 05h : bSlot 存在しない 07h : bPowerselect エラー (サポートされない) 08h : wLevelParameter エラー 0Ah : FI – DI ペアリングが無効またはサポートされていません 0Bh : TCCKTS パラメータ無効 0Ch : 保護時間サポートしない 0Dh : T = 0 WI 無効またはサポートされていません T = 1 BWI または CWI 無効またはサポートされていません 0Eh : 要求されたクロック停止のサポートが無効またはサポートされていません 0Fh : IFSC 大きさが無効またはサポートされていません 10h : NAD 数値が無効またはサポートされていません
00h	このコマンドをサポートできません。	

表18 : スロットビットエラーレジスタ (bmCommandStatus = 1)