HP Engage One Pro磁気ストライプ リーダー
ユーザー ガイド

M49143-291

RMN： HSN-NL02
© Copyright 2020 HP Development Company, L.P.
All rights reserved.AndroidはGoogle LLCの商標です。Linux®は、Linus Torvaldsの米国およびその他の国における登録商標です。MicrosoftおよびWindowsは、米国Microsoft Corporationの米国およびその他の国における商標または登録商標です。Enterprise LinuxおよびRed Hatは、Red Hat, Inc.の米国およびその他の国における商標です。

本書の内容は、将来予告なしに変更されることがあります。HP製品およびサービスに対する保証は、当該製品およびサービスに付属の保証規定に明示的に記載されているものに限られます。本書のいかなる内容も、当該保証に新たに保証を追加するものではありません。本書の内容につきましては万全を期しておりますが、本書の技術的あるいは校正上の誤り、省略に対して責任を負いかねますのでご了承ください。

初版：2020年12月
製品番号：M49143-291

製品についての注意事項

このユーザー ガイドでは、ほとんどのモデルに共通の機能について説明します。一部の機能は、お使いのコンピューターでは使用できない場合があります。

最新版のユーザー ガイドを確認するには、HPのサポートWebサイト、https://support.hp.com/jp-ja/ にアクセスし、説明に沿ってお使いの製品を探します。次に、**【マニュアル】**を選択します。

ソフトウェア条項

このコンピューターにプリインストールされている任意のソフトウェア製品をインストール、複製、ダウンロード、またはその他の方法で使用することによって、お客様はHP使用許諾契約（EULA）の条件に従うことに同意したものとみなされます。これらのライセンス条件に同意されない場合、未使用の完全な製品（付属品を含むハードウェアおよびソフトウェア）を14日以内に返品し、販売店の返金方針に従って返金を受けてください。

より詳しい情報が必要な場合またはコンピューターの代金の返金を要求する場合は、販売店に問い合わせてください。

目次

1. はじめに

HP Engage One Pro磁気ストライプ リーダー（MSR）では、1、2、または3トラックの磁気ストライプ情報を読み取ることができます。ホストに接続すると、磁気ストライプ リーダーにSPI（Serial Peripheral Interface）との完全な互換性が確保されます。未加工データおよびデコードされたデータは、SPI経由でホストに送信されます。また、ファームウェアもSPI経由でアップグレードできます。

MSRでは、暗号化されていないデータ出力と暗号化されたデータ出力の両方をサポートしています。暗号化が有効になっていない場合は、ホストで期待されるフォーマットと一致するように、デコードされたデータをプリアンブル、ポストアンブル、およびターミネータの各文字でフォーマットできます。

2. 仕様

全般	
カード速度	3 ~ 75 ips（7.6 ~ 190.5 cm/s）
電気的仕様	
電源	3.0 ~ 3.6 VDC
I/O電圧範囲	2.7 ~ 3.6 VDC
電流	
アクティブ状態の電源の電流	5
mA スタンバイ状態の電源の電流	0.03 mA
環境規格	
ESD	ヘッドへの+4 kVの放電
動作時温度	0 ~ 55℃
非動作時温度	-40 ~ 70℃
湿度	-10 ~ 90%（結露なし）
機械的仕様	
重量	5.7グラム
ケーブル長	125 +/- 6.4 mm

注：アナログ コンポーネントの復帰中に、いくつかのコンデンサーが充電され、復帰時の突入電流が5 μ秒以内に最大40 mAに達する可能性があります。

注2:チップの電源投入時に、内部レギュレーターは50 μ秒間に80 mAの電流を導入できます。

注3：MSRへの電源供給を個別に制御する機能を組み込むことをおすすめします。ファームウェアの更新手順中に、短い間（数秒間）デバイスの電源が切断された場合、ファームウェアのロードが失敗する可能性があります。ホストのソフトウェアでは、MSRの電源を入れ直して、再度デバイスを起動できます。その後、ファームウェアのロードを続行するには、約500ミリ秒以内にホストがユニットと通信する必要があります。

通常の操作では、ユニットの電源を切ることはおすすめしません。また、MSRデータを受信してから2秒以内に電源を切らないでください。

取り付け

MSRをHP Engage One Proに取り付けるには、以下の操作を行います。

1. MSRスロットを覆っているフレームを取り外します。
2. HP Engage One Proのシステム ケーブルをMSRに接続します。
3. MSRを挿入し、2本のネジで固定します。
4. MSRスロットを覆っているフレームを再度取り付けます。

3. SPI操作

このセクションでは、SPI（Serial Peripheral Interface）、SPIバス インターフェイスのタイミング、通信プロトコル、タイムアウト、およびデータ出力フォーマットについて説明します。以下の表に、SPIインターフェイスで使用される信号を示します。コネクタが8ピンのMolex 51021-0800であることに注意してください。

ピン番号	信号	説明
1	SPCK	シリアル クロック入力
2	MISO	送信側入力、受信側出力
3	MOSI	送信側出力、受信側入力
4	DAV	データ利用可能（出力）
5	NCS	チップ セレクト、アクティブ ロー
6	VIN	電圧入力
7	GND	ロジック グラウンド
8	ヘッド ケースGND	シャーシ グラウンド

SPIデータ送信

「シリアル ペリフェラル インターフェイス」（SPI）とは、2台のデバイス間でデータのシリアル交換を可能にするインターフェイスです。デバイスの一方を送信側、もう一方を受信側と呼びます。ホスト（送信側）では、SPIバス上でのデータ交換をトリガーするためのクロック信号（SPCK）を生成します。

各SPIクロック サイクル中に、データは双方向で同時に送信されます（全二重送信）。

- MOSI線では、送信側がビットを送信し、受信側がそれを読み取ります。
- MISO線では、受信側がビットを送信し、送信側がそれを読み取ります。

SPIバスでは、8ビットのデータ グループでデータを送信し、MSBからLSBにデータを一度に1ビットずつ送信します。（2バイト数のABの）バイトAおよびバイトBのビット送信の例は以下のようになります。

A（ビット7）A（ビット6）...A（ビット0）B（ビット7）B（ビット6）...B（ビット0）。

クロックの極性および位相

クロックの極性および位相には、データに関して4つの異なるオプションがあります。シリアル クロックの入力周波数は最大1 Mbpsになる可能性があります。

クロックの極性= 0の場合、クロックの基準値は0です。

クロックの位相= 0の場合、データはクロックの立ち上がりエッジ（ローからハイへの遷移）で読み取られ、立ち下がりエッジ（ハイからローへの遷移）で変更されます。

クロックの位相= 1の場合、データはクロックの立ち下がりエッジで読み取られ、立ち上がりエッジで変更されます。

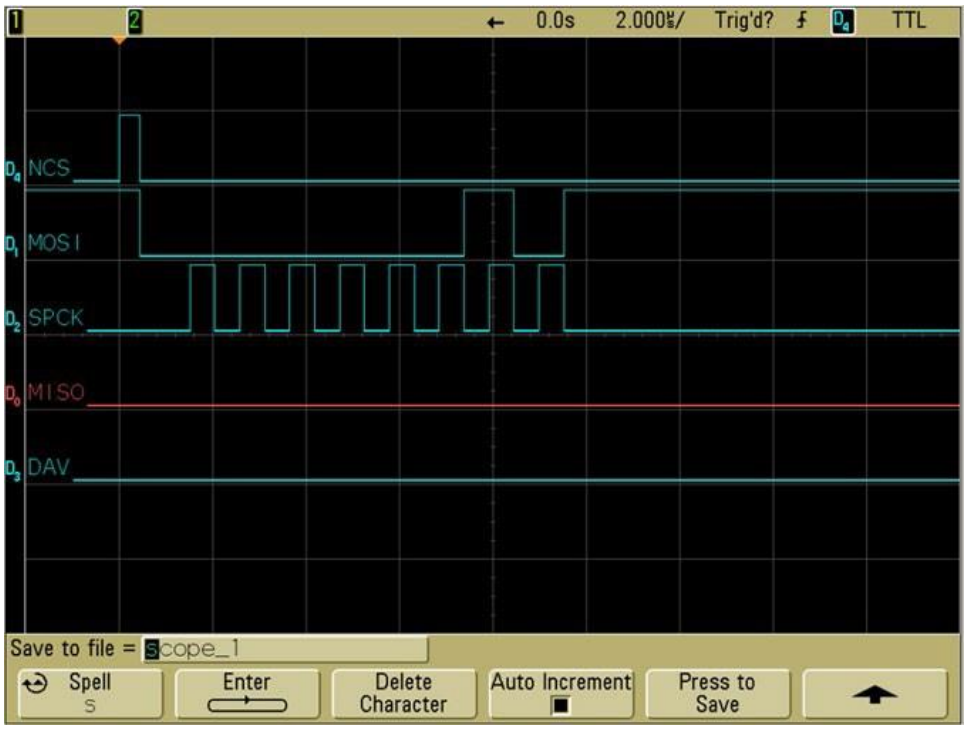
クロックの極性= 1の場合、クロックの基準値は1です。

クロックの位相= 0の場合、データはクロックの立ち下がりエッジで読み取られ、立ち上がりエッジで変更されます。

クロックの位相= 1の場合、データはクロックの立ち上がりエッジで読み取られ、立ち下がりエッジで変更されます。

デバイスからカード データを読み取るには信号が必要です。デバイスの初期設定では、クロックの位相= 0およびクロックの極性= 0を使用します。デバイス クロックの位相および極性のカスタム初期設定は、要求に応じて利用できます。

以下の画像は、クロックの極性= 0およびクロックの位相= 0を使用した通常のTM4 SPIファームウェアの例を示しています。データはクロックの立ち上がりエッジで読み取られ、立ち下がりエッジで変更されます。MOSI線では、ホストは00000010、つまり02h（0x02）のデータを送信します。



送信側入力、受信側出力（MISO）

MISO信号は、デバイスから送信されるシリアル データ出力です。また、ホストで受信されるデータ線でもあります。デバイスがアクティブでない場合（チップ セレクトがハイの場合）、MISOは高インピーダンスになります（切断されます）。デバイスの電源を入れ直したか、リセットした後、最長で1秒間は、MISO信号が不確定な状態になります。その間はこの信号を無視してください。

送信側出力、受信側入力（MOSI）

MOSI信号は、デバイスでのシリアル データ入力およびホストでのシリアル データ出力です。この信号は、ホスト（送信側）からデバイス（受信側）に送信されます。デバイス キーなどのデバイス パラメーターを設定して保存した後は、この信号が不要になる場合があります。使用されていない場合は、この信号をハイに設定します。

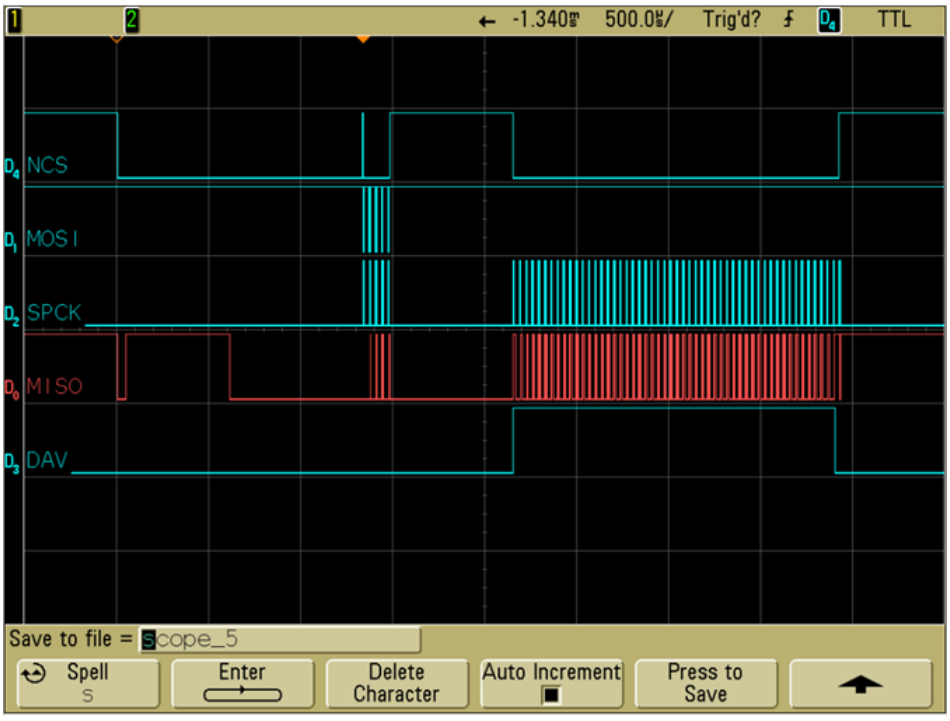
データ利用可能出力（DAV）

送信するデータがない場合、DAV信号はローです。DAV信号がハイの場合、出力可能なデータがあることを示します。そして、ホストはクロック信号を送信してそのデータを読み取ります。すべてのデータが送信された後、デバイスではDAV信号を再度ローに設定します。

この信号をホストに使用すると、デバイスに送信可能なデータがあるかどうかを確認できます。ただし、電源を入れ直したか、リセットした直後（最長で1秒間）は信号が不確定な状態になるため、無視してください。

DAV信号が使用されていない場合は、ホストでデバイスのポーリングを定期的に実行して、送信するデータがあるかどうかを確認する必要があります。ホストでは、MISOからカード データを取得するためにSCLを切り替える必要があります。IDLE以外の最初のバイトは、有効なカード データの開始を示しています。IDLEはFFです。

以下のグラフは、バージョンの確認コマンドのコマンドおよび応答を示しています。グラフに表示される最後の信号はDAV信号です。



コマンドが受信され、応答の準備が整った後は、DAVの設定が「high」となり、ホストで応答を受信できなくなります。応答が受信された後、DAVは「ロー」になり、送信するデータがこれ以上ないことを示します。

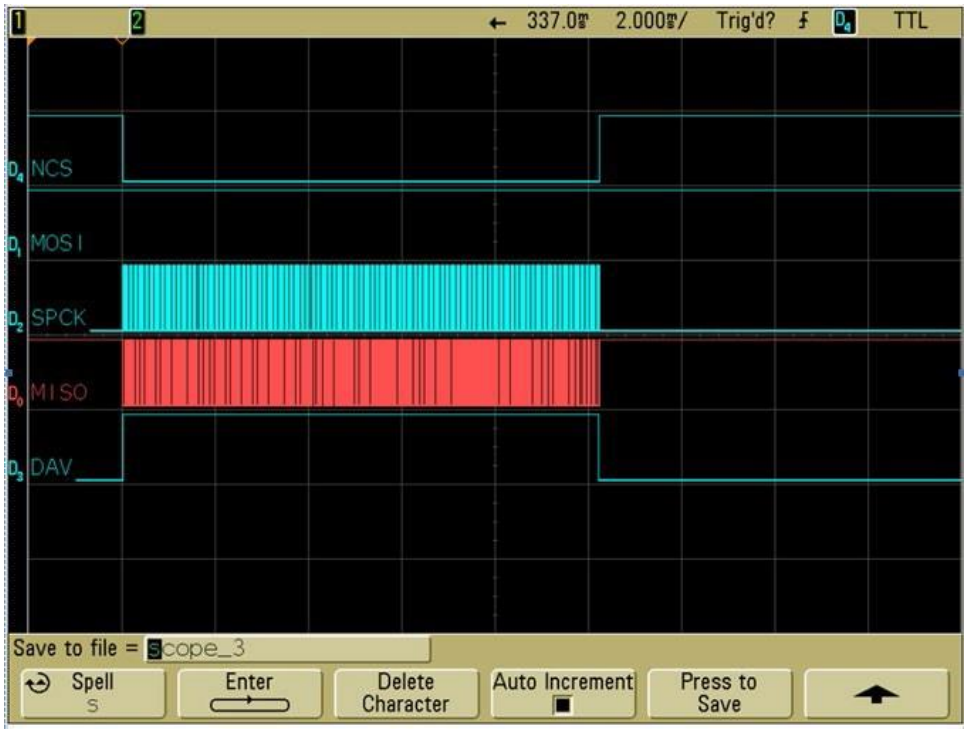
コマンドの受信後、通常は20ミリ秒以内に、応答の準備が整い、DAVが「ハイ」に設定されます。一部の特定のコマンドでは、遅延時間が長くなる場合があります。

応答の最後のバイトが送信された後、DAVはローに引き下げられます。ユーザーがDAVのステータスをポーリングして、利用可能なデータがあるかどうかを確認する場合は、100 µsのポーリング間隔を使用し、DAVがローのときにデータをすべて破棄することをおすすめします。

チップセレクト

SPIインターフェイスでは複数のSPIデバイスの接続が可能ですが、送信側ではNCS（チップ セレクト、アクティブ ロー）を使用してそれぞれのデバイスを選択します。デバイスは、NCSがローに引き下げられた後にのみSPCKやMOSIの信号に応答します。MSRに送信される各コマンドの最初のバイトでは、クロック線の前に1ミリ秒間NCSをローに引き下げる必要があります。MSRはアイドル状態のときは常にディープ スリープ モードであるため、MSRがスリープ モードから復帰できるようにするために、この1ミリ秒の遅延時間が必要です。

ユーザーがカードをスワイプするときは、遅延時間は必要ありません。MSRの出力の波形は以下のとおりです。



電圧入力およびグラウンド

VIN信号はデバイスの電源入力であり、動作範囲は3.0 ~ 3.6 VDCです。GND信号はロジック グラウンドです。ヘッド ケースGND信号とはシャーシ グラウンドであり、ヘッド ケースに接続されています。最適なESD保護が得られるように、この信号を接地に接続してください。

通信

送信するフレームがホストにある場合、ホストではNCS線をローに引き下げ、1ミリ秒待機してから、フレームをクロック アウトします。送信するフレームがデバイスにある場合、デバイスではデータ利用可能（DAV）信号を立ち上げ、ホストでNCS線がローに引き下げられ、フレームがクロック インされるまで待機します。ホストでは通常、IDLE文字をクロック アウトして、デバイスからフレームをクロック インします。デバイスでは通常、送信するものがないときに1つの送信バッファーにIDLEバイトをロードするため、DAV信号がアサートされた後にデバイスからクロック アウトされる最初のバイトは、有効なバイトではなくIDLEである可能性があります。その場合は、このバイトを破棄するだけです。

送信するフレームがデバイスにあるかどうかを検出するために、ホストではDAV信号を監視するか、オプションで、デバイスから最大2バイトを定期的にクロック インして、デバイスが有効なデータを送信したかどうかを確認できます。最初のバイトは、送信するものがデバイスに含まれる前にデバイスの送信バッファーにロードされたIDLEバイトである可能性があるため、1バイトではなく最大2バイトをクロック インしてください。ホストでは、クロック インする各バイトを調べて、それが有効なバイトであるかどうかを確認する必要があります。有効なバイトが見つかった場合、後続のバイトにはフレームが含まれます。

4. 設定

MSRは、お使いのアプリケーションに合わせて適切に設定する必要があります。設定により、リーダーをホスト システムで機能させることができます。プログラムすると、これらの設定はリーダーの不揮発性メモリに保存されます(そのため、電源の入れ直しの影響を受けません)。

TriMag IVでは、ACKは0x5Aです。

コマンド構造

MSRに送信されるコマンド

設定コマンド:

<STX><S>[<FuncID><Len><FuncData>...]<ETX><CheckSum>

読み取りステータス コマンド:

<STX><R><FuncID><ETX><CheckSum>

特別な機能コマンド:

<STX>[<FuncID><Len><FuncData>...]<ETX><CheckSum>

MSRからの応答

設定コマンド

ホスト　　MSR
　設定　　→
　コマンド
←　　>(OKの場合)
　または
←　<NAK>(エラーの場合)

読み取りステータス コマンド:

ホスト　　MSR

　読み取りステータス コマンド →
←　<ACK> および <応答>(OKの場合)
　または
←　<NAK>(エラーの場合)

特別な機能コマンド：

ホスト　　MSR

特別な機能 →
コマンド
← <ACK> および <応答>（OKの場合）
または
← <NAK>（エラーの場合）

このフォーマットは以下のように定義されています。

<STX>	02h
<S>	設定コマンドを示します。53h
<R>	読み取りステータス コマンドを示します。52h
<FuncID>	1バイトの機能IDで、を識別します
<Len>	次のデータの1バイトの長さカウント
<FuncData>	機能を表すデータ ブロック
<ETX>	03h
<CheckSum>	チェック サム：全体的なモジュラス2（排他的OR）の合計（<STX>から
<ACK>	06h
<NAK>	15h

通信のタイミング

TMIV搭載のMSRの起動までの時間は600ミリ秒です。リーダーがコマンドに応答するための通常の遅延時間は20ミリ秒です。リーダーが応答するための最大遅延時間が40ミリ秒になることもあります。したがって、2つのコマンド間で適切な遅延時間を維持するように注意する必要があります。

初期設定

MSRは、初期設定がすでにプログラムされた状態で工場から出荷されます。以降のセクションでは、初期設定を**太字**で示しています。

初期設定表については、付録Aを参照してください。

一般的な選択

この設定グループでは、MSRの基本的な動作パラメーターを定義します。

初期設定への変更

コマンド：<STX><S><18h><ETX><CheckSum>

このコマンドには<FuncData>がありません。すべてのグループのすべての設定が初期設定値に戻されます。

MSRの読み取り設定

MSRを有効または無効にします。リーダーが無効になっている場合は、ホストにデータが送信されません。

コマンド：<STX><S><1Ah><01h><MSRの読み取り設定><ETX><CheckSum>

MSRの読み取り設定：

- 0：無効
- 1：有効

デコード方式の設定

MSRでは、4種類のデコード方式をサポートできます。

コマンド：<STX><S><1Dh><01h><デコード方式の設定><ETX><CheckSum>

デコード方式の設定：

- 0：HPのモードで送信される、双方向での未加工データのデコード。
- 1：双方向でのデコード。暗号化機能が有効になっている場合、使用されるキー管理方式はDUKPTです。
- 2：エンコードの方向にストライプをヘッドに沿って移動。暗号化機能が有効になっている場合、使用されるキー管理方式はDUKPTです。
- 3：エンコードの方向とは逆にストライプをヘッドに沿って移動。暗号化機能が有効になっている場合、使用されるキー管理方式はDUKPTです。
- 4：他のモードで送信される、双方向での未加工データのデコード。暗号化機能が有効になっている場合、使用されるキー管理方式は固定キーです。

双方向方式では、ユーザーはどちらの方向にカードをスワイプしても、磁気ストライプ上のエンコードされたデータを読み取ることができます。それ以外の場合、カードを読み取るには、指定された1方向にのみカードをスワイプします。未加工のデコードでは、カードの磁気データを1文字あたり4ビットのグループで送信するだけです。

ヘッドでは、各トラックの最初のバイトの最上位ビットから読み取りを開始します。最初の1ビットが検出されると、データの収集が開始されます。トラックに磁気データがあるかどうかを確認する以外のチェックは行われません。

Samsung Payのエンコードまたはデコード

Samsung Payインタラクションには、トラック デコードに関する特別な考慮事項が適用されます。Samsung Pay/MST（LoopPay）では、磁気信号を磁気ヘッドに送信します。そのため、MCUではすべてのトラックについて同一の磁気信号を受信する可能性があります。ただし、Samsung Payデバイスではトラック1およびトラック2のデータを連続して送信するため、それらのトラックの曖昧さを解消することができます。

読み取りデバイスで複数のトラックについて同一のMSRデータを受信した場合、カード データがISO 7ビットでエンコードされていれば、MSR処理ではトラック2およびトラック3のデータは無視され、トラック1データとして扱われます。データが5ビットでエンコードされている場合は、トラック2のデータとしてのみ受信されます。

MSRでABA、IATA、またはISO 4909に対応する単一のトラック データを受信したが、想定されたトラック内にない場合は、トラック データを誤ったタイプとしてキャプチャしないように、そのデータは無視されます。プロセッサでは、トラック間でのデータの移動は行いません。

設定の確認

コマンド：<STX><R><1Fh><ETX><CheckSum>

このコマンドには<FuncData>がありません。これにより、設定の確認コマンドがアクティブになります。MSRでは<ACK>および<Response>を送り返します。

<Response>の形式：

現在の設定データ ブロックは、以下のように、多くの機能設定ブロック<FuncSETBLOCK>のコレクションです。

<STX><FuncSETBLOCK1>...<FuncSETBLOCKn><ETX><CheckSum>

各機能設定ブロック<FuncSETBLOCK>には、以下の形式があります。

<FuncID><Len><FuncData>

この形式は以下のように定義されています。

- <FuncID>は、機能の設定を識別する1バイトです。

- <Len>は、以下の機能設定ブロック<FuncData>の1バイトの長さカウントです。
- <FuncData>は、この機能の現在の設定です。この機能の送信コマンドと同じ形式です。
- <FuncSETBLOCK>は、その機能ID<FuncID>の順序になります。

ファームウェアのバージョンの確認

コマンド：<STX><R><22h><ETX><CheckSum>

このコマンドでは、デバイス ファームウェアのバージョンを取得します。

シリアル番号の確認

コマンド：<STX><R><4Eh><ETX><CheckSum>

このコマンドでは、デバイス シリアル番号を取得します。

メッセージ フォーマットの選択（セキュリティ レベル1および2のみ）

ターミネータ設定

一部のアプリケーションでは、ターミネータ文字を使用して一連のデータを終了します。

コマンド：<STX><S><21h><01h><ターミネータ設定><ETX><CheckSum>

<ターミネータ設定>：任意の1文字で、00hは「なし」を示します。初期設定は**CR**（0Dh）です。

プリアンブル設定

一連のデータの先頭に文字を追加できます。特定の読み取り位置を識別するための特殊文字（受信側ホストで想定されるメッセージ ヘッダーをフォーマットするため）またはその他の文字列を指定できます。最大15のASCII文字を定義できます。

コマンド：<STX><S><D2h><Len><Preamble><ETX><CheckSum>

この形式は以下のように定義されています。

- <Len> =プリアンブル文字列のバイト数
- <Preamble> = {文字列の長さ}{文字列}

注：文字列の長さは1バイトで、最大15<0Fh>です。

ポストアンブル設定

ポストアンブルは、データ文字列の末尾（ターミネータ文字の後）に追加されることを除いて、プリアンブルと同じ目的で使用されます。

コマンド： <STX><S><D3h><Len><Postamble><ETX><CheckSum>

この形式は以下のように定義されています。

- <Len> =ポストアンブル文字列のバイト数
- <Postamble> = {文字列の長さ}{文字列}

注： 文字列の長さは1バイトで、最大15<0Fh>です。

トラックnのプレフィックス設定

トラック データの先頭に文字を追加できます。受信側ホストへの特定のトラックを識別するための特殊文字、またはその他の文字列を指定できます。最大6のASCII文字を定義できます。

コマンド： <STX><S><n><Len><Prefix><ETX><CheckSum>

この形式は以下のように定義されています。

- <n> = トラック1の場合は34h、トラック2の場合は35h、トラック3の場合は36h
- <Len> =プレフィックス文字列のバイト数
- <Prefix> = {文字列の長さ}{文字列}

注： 文字列の長さは1バイトで、最大6です。

トラックnのサフィックス設定

トラック データの末尾に文字を追加できます。受信側ホストへの特定のトラックを識別するための特殊文字、またはその他の文字列を指定できます。最大6のASCII文字を定義できます。

コマンド： <STX><S><n><Len><Suffix><ETX><CheckSum>

この形式は以下のように定義されています。

<n> = トラック1の場合は37h、トラック2の場合は38h、トラック3の場合は39h

<Len> =サフィックス文字列のバイト数

<Suffix> = {文字列の長さ}{文字列}

注：文字列の長さは1バイトで、最大6です。

磁気トラックの選択（セキュリティ レベル1および2のみ）

磁気ストライプには、最大3トラックのエンコードされたデータがあります。このオプションでは、読み取られてデコードされるトラックを選択します。

コマンド：<STX><S><13h><01h><トラックの選択設定><ETX><CheckSum>

トラックの選択設定：

- 0：任意のトラック
- 1：トラック1のみが必要
- 2：トラック2のみが必要
- 3：トラック1およびトラック2が必要
- 4：トラック3のみが必要
- 5：トラック1およびトラック3が必要
- 6：トラック2およびトラック3が必要
- 7：3つのトラックすべてが必要
- 8：任意のトラック1および2
- 9：任意のトラック2および3

注：必要な複数のトラックのどれかの読み取りが何らかの理由で失敗した場合、どのトラックのデータも送信されません。

トラック セパレーターの選択

このオプションを使用すると、ユーザーは、マルチトラック リーダーによってデコードされたデータを区切るために使用する文字を選択できます。

コマンド：<STX><S><17h><01h><Track_Separator><ETX><CheckSum>

<Track_Separator>は、1つのASCII文字です。初期設定値は**CR**で、0hはトラック セパレーターがないことを意味します。

開始/終了符号およびトラック2のアカウント番号のみ

MSRは、開始/終了符号を送信するかしないか、およびトラック2のアカウント番号のみを送信するか、トラック2上のすべてのエンコードされたデータを送信するかを設定できます。(トラック2のアカウント番号設定は、トラック1およびトラック3の出力には影響しません。)

コマンド：<STX><S><19h><01h><SendOption><ETX><CheckSum> SendOption：

- 0：開始/終了符号を送信せず、トラック2上のすべてのデータを送信します
- 1：開始/終了符号を送信し、トラック2上のすべてのデータを送信します
- 2：開始/終了符号を送信せず、トラック2上のアカウント番号を送信します
- 3：開始/終了符号を送信し、トラック2上のアカウント番号を送信します

5. セキュリティ設定

キー管理タイプの選択

コマンド：<STX><S><58h><01h><キー管理タイプ><ETX><CheckSum>

使用可能なキー管理タイプは以下のとおりです。

- 0：固定キー管理
- 1：DUKPTキー管理

外部認証コマンド（固定キーのみ）

セキュリティ関連コマンドを実行する前に、使用されているデバイス キーが正しいことを確認するための認証プロセスが必要です。たとえば、暗号化が有効/無効になる場合、またはデバイス キーが変更される場合は通常、認証が必要です。認証プロセスが正常に完了したら、デバイスが再起動されるまで、同じプロセスを再度実行する必要はありません。

ホストではまず、TDESアルゴリズムでランダムな8バイト データを暗号化して生成されたデータ ブロックを取得します。

次に、現在のデバイス キーを使用してそのデータ ブロックをTDESアルゴリズムで復号化します。

ホストでは外部認証コマンドを開始して、復号化された8バイトのランダム データを検証します

デバイスでは、データが生成されたランダム データと一致するかどうかを確認します。データが同じである場合、認証プロセスは正常に行われています。失敗した場合、ホストでは、セキュリティ関連の機能を変更する前に、成功するまで認証プロセスを繰り返し実行する必要があります。

暗号化されたチャレンジ コマンドの取得

ホストからデバイス：

コマンド：<STX><R><74h><ETX><CheckSum>

デバイスからホスト：

コマンド：<ACK><STX><TDESで暗号化された8バイトのランダム データ><ETX><CheckSum>（成功）

<NAK>（失敗）

ホストからデバイスへの外部認証コマンドの送信

コマンド：<STX><S><74h><08h><元の8バイトのランダム データ><ETX><CheckSum>

デバイスからホスト：

<ACK>（成功）

<NAK>（失敗）

暗号化設定

ID TECHプロトコルのMSR暗号化出力を有効または無効にします。暗号化が無効になっている場合は、元のデータがホストに送信されます。有効になっている場合は、暗号化されたデータがホストに送信されます。

コマンド：<STX><S><4Ch><01h><暗号化設定><ETX><CheckSum>

使用可能な暗号化設定は以下のとおりです。

- 0：暗号化が無効
- 1：TDES暗号化を有効にする
- 2：AES暗号化を有効にする

KSNの確認（DUKPTキー管理のみ）

コマンド：<STX><R><51h><ETX><CheckSum>

このコマンドでは、DUKPTキー シリアル番号およびカウンターを取得します。

セキュリティ レベルの確認

コマンド：<STX><R><7Eh><ETX><CheckSum>

このコマンドでは、現在のセキュリティ レベルを取得します。

外部データの暗号化コマンド

このコマンドでは、MSRに渡されるデータを暗号化し、暗号化されたデータをホストに送り返します。このコマンドは、セキュリティ レベルが3または4に設定されている場合に有効です。

ホストからデバイス：

コマンド：<STX><41h><Length<暗号化するデータ><ETX><CheckSum>

この形式は以下のように定義されています。

<Length>は、<暗号化するデータ>の2バイトの長さ（16進数）であり、<Length_L>および<Length_H>で表されます。

デバイスからホスト：

コマンド：<ACK><STX><長さ><暗号化されたデータ>[セッションID]<KSN><ETX><LRC>（成功）または<NAK>（失敗）

この形式は以下のように定義されています。

<長さ>は、<暗号化されたデータ>[セッションID]<KSN>の2バイトの長さ（16進数）であり、<Length_L>および<Length_H>で表されます。

[セッションID]は、セキュリティ レベル4でのみ使用され、暗号化されたデータの一部になっています。セキュリティ レベル3では、このフィールドにデータはありません。

<KSN>は10バイトの文字列であり、固定キー管理の場合は、KSNではなくシリアル番号に2バイトのNULL文字を加えたものを使用します。

応答が正常に行われるたびに、KSNが自動的に増分されます。

デコードされたデータの暗号化出力

暗号化機能

カードをリーダーに通すと、トラック データが固定キー管理またはDUKPT（Derived Unique Key Per Transaction）キー管理を使用して、TDES（トリプル データ暗号化アルゴリズム、別名トリプルDES）またはAES（Advanced Encryption Standard）で暗号化されます。DUKPTキー管理では、ベース派生キーを使用して、初回暗号化キー（IPEK）を生成するキー シリアル番号を暗号化します。IPEKは展開前にMSRに挿入されます。各トランザクションの後、この暗号化キーはDUKPTアルゴリズムに従って変更され、各トランザクションで一意のキーが使用されるようになります。したがって、リプレイ攻撃に対する防御手段として、データはトランザクションごとに異なる暗号化キーで暗号化されます。DUKPTは、ANSI X9.24-1：2009で説明されています。詳しくは、その仕様を参照してください。

セキュリティ関連の機能ID

セキュリティ関連の機能IDは、以下の表に記載されています。それらの機能については、他のセクションで説明されています。

文字	16進値	説明
PrePANID	49	PANの最初のN桁。クリア データにすることができます
PostPANID	4A	PANの最後のM桁。クリア データにすることができます
MaskCharID	4B	PANをマスクするために使用される文字
EncryptionID	4C	セキュリティ アルゴリズム
SecurityLevelID	7E	セキュリティ レベル（読み取り専用）
デバイス シリアル番号ID	4E	デバイス シリアル番号（1回だけ書き込み可能です。その後は、読み取りのみ可能です）
DisplayExpirationDataID	50	期限切れのデータをマスク データまたはクリア データとして表示します
KSNおよびカウンターID	51	キー シリアル番号および暗号化カウンターを確認します
セッションID	54	現在のセッションIDを設定します
キー管理タイプ ID	58	キー管理タイプを選択します

以下の例は、これらの新しい機能の考えられる設定を示したものです。

文字	初期設定	説明
PrePANID	04h	00h ~ 06h PANの最初のテキストをクリア テキストにできます コマンド形式： 02 53 49 01 04 03 LRC
PostPANID	04h	00h ~ 04h PANの最後のテキストをクリア テキストにできます コマンド形式： 02 53 4A 01 04 03 LRC
MaskCharID	‘*’	20h ~ 7Eh コマンド形式： 02 53 4B 01 3A 03 LRC
DisplayExpirationDataID	‘0’	0：期限切れデータをマスク データとして表示します 1：期限切れデータをクリア データとして表示します
EncryptionID	‘0’	0：クリア テキスト 1：トリプルDES 2：AES コマンド形式：02 53 4C 01 31 03 LRC
SecurityLevelID	‘1’	0 ~ 3 コマンド形式： 02 52 7E 03 LRC
デバイス シリアル番号ID	00, 00, 00, 00, 00, 00, 00, 00,	10バイトの番号：コマンド形式： シリアル番号の設定： 02 53 01 4E 09 08 37 36 35 34 33

	00, 00	32 31 30 03 LRC シリアル番号の取得： 02 52 4E 03 LRC
KSNおよびカウンターID	00, 00, 00, 00, 00, 00, 00, 00, 00, 00	このフィールドには、左端の59ビットに初回キー シリアル番号が含まれ、 右端の21ビットに暗号化カウンターの値が含まれます。DUKPTのKSNおよびカウンターの取得： 02 52 51 03 LRC
セッションID	00, 00, 00, 00, 00, 00, 00, 00	このセッションIDは、16進データを含む8バイト文字列です。このフィールドは、現在のトランザクションを一意に識別するためにホストによって使用されます。その主な目的は、リプレイを防ぐことです。セキュリティレベル4でのみ使用されます（サポートされていません）。カードが読み取られた後、セッションIDはカード データとともに暗号化され、トランザクション メッセージの一部として提供されます。これのクリアテキストバージョンは送信されません。 新しいセッションIDは、以下のどれかが発生するまで有効です。 別のセッションIDの設定コマンドが受信される リーダーの電源がオフになる リーダーがサスペンド モードになる
キー管理タイプID	‘1’	初期設定では、固定キー管理になっています。 0：固定キー 1：DUKPTキー

セキュリティ管理

このMSRは、安全なリーダーとなるよう意図されています。セキュリティ機能には以下のようなものがあります。

- デバイス シリアル番号を含めることができます
- すべてのバンク カードのトラック1およびトラック2のデータを暗号化できます
- カード保有者の名前やPANの一部を含むクリア テキストの確認データをマスクされたトラック データの一部として提供します
- 期限切れデータをオプションで表示します
- セキュリティ レベルを設定できます

このリーダーは、設定可能なセキュリティ設定を備えています。暗号化を有効にする前に、キー シリアル番号（KSN）とベース派生キー（BDK）をロードする必要があります。その後、暗号化されたトランザクションを実行できます。これらのキーは、認定されたキー インジェクション機能(ID TECHなど)によって挿入される必要があります。キー インジェクション サービスについて詳しくは、ID TECHに問い合わせてください。

レベル0

セキュリティ レベル0は、すべてのDUKPTキーが使用された特殊なケースであり、DUKPTキーが不足すると自動的に設定されます。DUKPTキーの供給は実質的に100万個です。つまり、スワイプごとに新しいキーが1つ生成され、最大100万回のカード スワイプに備えることができます。この制限に達したら、トランザクションをさらに実行する前に、キー インジェクションを再度行う必要があります。

レベル1

初期設定では、工場出荷時のリーダーは、このセキュリティ レベルを持つように設定されています。暗号化プロセスはなく、デコードされたデータとともに送信されるキー シリアル番号もありません。このリーダーは非暗号化リーダーとして機能し、デコードされたトラック データは初期設定モードで送信されます。

レベル2

キー シリアル番号とベース派生キーが挿入されていますが、暗号化プロセスはまだアクティブ化されていません。このリーダーではデコードされたトラック データを初期設定のフォーマットで送信します。暗号化タイプをTDESおよびAESに設定すると、リーダーはセキュリティ レベル3に変更されます。

レベル3

キー シリアル番号とベース派生キーの両方が挿入され、暗号化モードがオンになります。ペイメント カードの場合、暗号化されたデータとマスクされたクリアテキスト データの両方が送信されます。(ユーザーはPAN領域のデータ マスキングを選択できます。暗号化されたデータのフォーマットは変更できません。) ハッシュ データを送信するかどうか、およびカードの有効期限を公開するかどうかを選択できます。暗号化が有効になっている場合、レベル3は初期設定のセキュリティ レベルです。

6. 暗号化管理

暗号化されたスワイプ読み取りは、データ暗号化のTDESおよびAES暗号化標準に対応しています。コマンドを使用して暗号化を有効にすることができます。TDESは初期設定です。

リーダーがセキュリティ レベル3以上の場合、暗号化されたフィールドでは、元のデータがTDES/AES CBCモードを使用して暗号化され、すべてのバイナリの初期化ベクトルがゼロに設定され、暗号化キーが現在のDUKPT KSNに関連付けられます。

カード フォーマットの確認

ISO/ABA（American Banking Association）カードのエンコード方式

トラック1は7ビットでのエンコードです。トラック1は7ビットでのエンコードです。トラック2は5ビットでのエンコードです。トラック3は5ビットでのエンコードです。トラック1は7ビットでのエンコードです。トラック2は5ビットでのエンコードです。トラック2は5ビットでのエンコードです。

その他の確認：

トラック1の2番目のバイトはBです。

トラック2には「=」が1つだけあり、「=」の位置は13 ~ 20文字目の間です。トラック2の全体の長さが21文字を超えるようにしてください。

AAMVA（American Association of Motor Vehicle Administration）カードのエンコード方式

トラック1は7ビットでのエンコードです。トラック2は5ビットでのエンコードです。トラック3は7ビットでのエンコードです。

7. その他（顧客カード）

MSRデータのマスキング

暗号化が必要なカードの場合、暗号化されたデータとマスクされたクリア テキスト データの組み合わせが送信されます。

マスクされた領域

マスクされた各トラックのデータ フォーマットはASCIIです。

クリア データには、開始および終了符号、セパレーター、PANの最初のN桁、最後のM桁、カード保有者の名前が含まれます（Track1の場合）。

残りの文字は、マスク文字を使用してマスクする必要があります。

`Set PrePANClrData (N), PostPANClrData (M), MaskChar (マスク文字)`

NおよびMは設定可能であり、初期設定は最初の4桁および最後の4桁です。これらは、現在のPCI制約の要件（最大でN 6、M 4）に準拠しています。

マスク文字の初期設定は「*」です。

Set PrePANClrDataID (N)

パラメーターの範囲は00h ~ 06hで、初期設定は04hです。

Set PostPANClrDataID (M)

パラメーターの範囲は00h ~ 04hで、初期設定は04hです。

MaskCharID（マスク文字）

パラメーターの範囲は20h ~ 7Ehで、初期設定は2Ahです。

DisplayExpirationDataID

パラメーターの範囲は0 ~ 1で、初期設定は0です。

レベル1および2のデータ出力フォーマット

磁気トラックのデコードされたデータの基本フォーマット

Track1: <SS1><T1 Data><ES><トラック セパレーター> Track2: <SS2><T2 Data><ES><トラック セパレーター> Track3: <SS3><T3 Data><ES><ターミネータ>

このフォーマットは以下のように定義されています。

SS1（トラック1の開始符号）= % SS2（トラック2の開始符号）= ;

SS3（トラック3の開始符号）= ;（ISOの場合）、%（AAMVAの場合） ES（全トラックの終了符号）= ?

トラック セパレーター = キャリッジ リターン ターミネータ = キャリッジ リターン 言語：英語（米国）

磁気トラックの未加工データの基本フォーマット

Track1: <01><T1の未加工データ><CR> Track2: <02><T2の未加工データ><CR> Track3: <03><T3の未加工データ><CR>

このフォーマットは以下のように定義されています。T1の未加工データ、T2の未加工データ、T3の未加工データの長さは、各フィールドで0x60です。元のデータの長さが0x60に達しない場合は、0を埋め込みます。

言語：英語（米国）

定義

開始または終了符号：最初のデータ文字（開始）の前および最後のデータ文字（終了）の後に置かれるエンコード形式の文字。それぞれ、データの開始と終了を示します。

トラック セパレーター：データのトラックを区切る指定文字。

ターミネータ：カードの読み取りを区切るために、データの最後のトラックの終わりに置かれる指定文字。

DUKPTレベル3のデータ出力拡張フォーマット

ISOカードの場合、マスクされたクリア データと暗号化されたデータの両方が送信されます。マスクされていないクリア データは送信されません。他のカードの場合、クリア データのみが送信されます。

このモードが使用されるのは、すべてのトラックを暗号化する必要がある場合、暗号化されたOPOSサポートが必要な場合、トラックを個別に暗号化する必要がある場合、タイプ0以外のカード（ABAバンク カード）を暗号化する必要がある場合、またはトラック3を暗号化する必要がある場合です。このフォーマットは標準の暗号化フォーマットですが、まだ初期設定の暗号化フォーマットにはなっていません。

カード データは以下のフォーマットで送信されます

<STX><LenL><LenH><カード データ><CheckLRC><CheckSum><ETX>

値	説明
0	STX
1	データ長の下位バイト
2	データ長の上位バイト
3	カードのエンコード タイプ1
4	トラック1 ~ 3のステータス2
5	トラック1のデータ長
6	トラック2のデータ長
7	トラック3のデータ長
8	クリアまたはマスク データの送信ステータス3
9	暗号化またはハッシュ データの送信ステータス4
10	トラック1のクリアまたはマスク データ
	トラック2のクリアまたはマスク データ

トラック3のクリアまたはマスク データ

トラック1の暗号化データ

トラック2の暗号化データ

トラック3の暗号化データ

レベル4のセッションID情報（レベル4は使用できません）

トラック1のハッシュ（各20バイト）（暗号化され、トラック1のハッシュが許可されている場合）

トラック2のハッシュ（各20バイト）（暗号化され、トラック2のハッシュが許可されている場合）

トラック3のハッシュ（各20バイト）（暗号化され、トラック3のハッシュが許可されている場合）KSN（10バイト）

`CheckLRC CheckSum ETX`

このフォーマットは以下のように定義されています。

- <STX> = 02h
- <ETX> = 03h

実際の例については、付録Fを参照してください。

データ長のバイト

LenL：データの全長、下位ビット LenH：データの全長、上位ビット

カードのエンコードタイプ

値	エンコードタイプの説明
80	ISO 7813/ISO 4909/ABAフォーマット
81	AAMVAフォーマット
83	その他
84	未加工。デコードされていないフォーマット

すべてのトラックが暗号化され、マスク データは送信されません。

各トラックの前にトラック インジケーター「01」、「02」、または「03」はありません。

85	JIS II。一部の製品でのみサポートされています
86	JIS I。一部の製品でのみサポートされています
87	JIS II。安全なキーおよび安全なMIR
91	非接触型Visa（カーネル1）
92	非接触型SenderCard
93	非接触型Visa（カーネル3）
94	非接触型American Express
95	非接触型JCB
96	非接触型Discover
97	非接触型UnionPay
90	その他の非接触型
C0	手動データ入力拡張モード（ABAのトラック2と同様）

トラックのステータス

MSRのサンプリングおよびデコードのステータス

MB　　LB

B7	B6	B5	B4	B3	B2	B1	B0

B0　1：トラック1のデコードに成功（0：トラック1のデコードに失敗）

B1　1：トラック2のデコードに成功（0：トラック2のデコードに失敗）

B2　1：トラック3のデコードに成功（0：トラック3のデコードに失敗）

B3　1：トラック1のサンプリング データあり（0：トラック1のサンプリング データなし）

B4　1：トラック2のサンプリング データあり（0：トラック2のサンプリング データなし）

B5　1：トラック3のサンプリング データあり（0：トラック3のサンプリング データなし）

B6　0：将来の使用のために予約済み

B7　0：将来の使用のために予約済み

トラックのデータ長

この1バイト値は、それぞれのトラックのマスクされたデータ フィールドのバイト数を示します。ISO 7813およびISO 4909準拠の金融取引カードの場合：

トラック1の最大長は79文字の英数字です。トラック2の最大長は40桁の数字です。トラック3の最大長は107桁の数字です。

クリアまたはマスク データの送信ステータス

ビット0　1：トラック1のクリアまたはマスク データあり
ビット1　1：トラック2のクリアまたはマスク データあり
ビット2　1：トラック3のクリアまたはマスク データあり
ビット3　1：固定キー

ビット4　0：TDES

ビット5　0：ICを使用するための要件なし

ビット6　1：PIN暗号化キー

ビット7　1：シリアル番号あり

暗号化されたハッシュ データの送信ステータス

ビット0　1：トラック1の暗号化データあり
ビット1　1：トラック2の暗号化データあり
ビット2　1：トラック3の暗号化データあり
ビット3　1：トラック1のハッシュ データあり
ビット4　1：トラック2のハッシュ データあり
ビット5　1：トラック3のハッシュ データあり
ビット6　1：セッションIDあり
ビット7　1：KSNあり

マスクされたトラック データ

MaskCharIDでマスクされたトラック データです（初期設定は「*」）。最初のPrePANID（バイナリの場合は最大6、初期設定は4）および最後のPostPANID（最大4、初期設定は4）の文字をクリア（非暗号化）にすることができます。

暗号化されたトラック データ

このフィールドは暗号化されたトラック データであり、初期ベクトルが0のTDES-CBCまたはAES-CBCを使用します。元のデータが8バイトの倍数（TDESの場合）または16バイトの倍数（AESの場合）でない場合は、リーダーによってデータに0が正しく埋め込まれます。

キー管理方式は、DUKPTまたは固定キーです。DUKPTの場合、データの暗号化に使用されるキーはデータ キーと呼ばれます。データ キーを生成するには、最初にDUKPT派生キーと0000000000FF0000との排他的論理和を取って、結果の中間バリアント キーを取得します。次に、16バイトのバリアント全体を使用して中間バリアント キーの左側をキーとしてTDESで暗号化します。

キーの右側でも同じステップを実行した後、2つのキー部分を組み合わせてデータ キーを作成します。

ハッシュされたトラック データ

MSRリーダーでは、SHA-1を使用して、トラック1、トラック2、およびトラック3の暗号化されていないデータのハッシュ データを生成します。その長さは、トラックごとに20バイトです。これは、2つの目的を念頭に置いて提供されます。1つは、ホストで、このフィールドを復号化されたトラック データのSHA-1ハッシュと比較してデータの整合性を確保し、データ送信での予期しないノイズの発生を防ぐためです。もう1つは、ホストで、カード保有者の機密性の高いデータを保持しなくても、将来使用するためのカード データのトークンを保存できるようにするためです。このトークンは、保存されているハッシュ データと比較して、それらが同じカードからのものであるかどうかを判断するために使用できます。

暗号化出力フォーマットの設定

コマンド： 53 85 01 <暗号化フォーマット>

暗号化フォーマットのオプションは次のとおりです。

0：サポートされなくなりました

1：拡張暗号化フォーマット

暗号化オプションの設定（拡張暗号化フォーマットの場合のみ）

コマンド： 53 84 01 <暗号化オプション>

暗号化オプションは次のとおりです（初期設定は08h）：

ビット0： 1：トラック1が強制的に暗号化されます
ビット1 1：トラック2が強制的に暗号化されます
ビット2 1：トラック3が強制的に暗号化されます
ビット3 1：カード タイプが0の場合にトラック3が強制的に暗号化されます

注：

- 強制暗号化が設定されている場合、カード タイプに関係なく、このトラックは常に暗号化されます。クリア テキストまたはマスク テキストは送信されません。
- 拡張暗号化フォーマットの場合に限り、各トラックは個別に暗号化されます。暗号化されたデータの長さは、8または16バイトに切り上げられます。
- 強制暗号化が設定されていない場合、データは元の暗号化フォーマットで暗号化されます。つまり、タイプ0のカード（ABAバンク カード）のトラック1およびトラック2のみが暗号化されます。

ハッシュ オプションの設定：

コマンド： 53 5C 01 <ハッシュ オプション>

ハッシュ オプションは次のとおりです（初期設定は7）。

ビット0 1：データが暗号化されている場合にトラック1のハッシュが送信されます

ビット1　1：データが暗号化されている場合にトラック2のハッシュが送信されます
ビット2　1：データが暗号化されている場合にトラック3のハッシュが送信されます

マスク オプション設定（拡張暗号化フォーマットの場合のみ）

コマンド： 53 86 01 <マスク オプション> マスク オプション：

初期設定は**0x07**です。

ビット0：1：暗号化されている場合にトラック1のマスク データの送信が許可されます

ビット1：1：暗号化されている場合にトラック2のマスク データの送信が許可されます

ビット2：1：暗号化されている場合にトラック3のマスク データの送信が許可されます

マスク オプション ビットが設定されていると、データが暗号化されている（ただし、強制的に暗号化されていない）場合にマスク データが送信されます。

マスク オプションが設定されていない場合、マスク データは同じ条件下で送信されません。

カードのエンコード タイプ

カード タイプは、拡張暗号化フォーマットの場合は8x、元の暗号化フォーマットの場合は0xです。

値	エンコード タイプ
00h/80h	ISO/ABAフォーマット
01h/81h	AAMVAフォーマット
03h/83h	その他
04h/84h	未加工。デコードされていないフォーマット

タイプ04または84の未加工データ フォーマットの場合、すべてのトラックが暗号化され、マスク データは送信されません。各トラックの前にトラック インジケーター「01」、「02」、または「03」はありません。非暗号化モードでは、トラック インジケーター「01」、「02」、および「03」は引き続き存在します。

トラック1～3のステータス バイト

フィールド4：

ビット0　1：トラック1のデコードされたデータあり
ビット1　1：トラック2のデコードされたデータあり
ビット2　1：トラック3のデコードされたデータあり
ビット3　1：トラック1のサンプリング データあり
ビット4　1：トラック2のサンプリング データあり
ビット5　1：トラック3のサンプリング データあり
ビット6　1：フィールド10「オプションのバイトの長さ」あり（0：フィールド10なし）

クリアまたはマスク データの送信ステータス

フィールド8（クリアまたはマスク データの送信ステータス）およびフィールド9（暗号化またはハッシュ データの送信ステータス）は、拡張暗号化フォーマットでのみ送信されます。

フィールド8：クリアまたはマスク データの送信ステータス バイト：

ビット0　1：トラック1のクリアまたはマスク データあり
ビット1　1：トラック2のクリアまたはマスク データあり
ビット2　1：トラック3のクリアまたはマスク データあり
ビット3　固定キーの場合は1
ビット4　0：TDES

ビット5　0：ICを使用するための要件なし（サービス コードの1桁目が2または6とは異なります）

ビット6　1：PIN暗号化キー

ビット7　1：シリアル番号あり

暗号化またはハッシュ データの送信ステータス

フィールド9：暗号化されたデータの送信ステータス

ビット0　1：トラック1の暗号化データあり
ビット1　1：トラック2の暗号化データあり
ビット2　1：トラック3の暗号化データあり
ビット3　1：トラック1のハッシュ データあり
ビット4　1：トラック2のハッシュ データあり
ビット5　1：トラック3のハッシュ データあり
ビット6　1：セッションIDあり
ビット7　1：KSNあり

固定キー管理のデータ出力拡張フォーマット

4.14.10 DUKPTレベル3データ出力拡張フォーマットと同じですが、<KSN>だけは<デバイス シリアル番号>に2つのNULLバイトを加えたものに変更します。

8. 付録A：初期設定表

MSRの読み取り	有効
デコード方式	双方向スワイプ デコード モード
トラック セパレーター設定	CR
ターミネータ設定	CR
プリアンブル設定	なし
ポストアンブル設定	なし
トラックの選択設定	任意のトラック
符号およびT2のアカウント番号	符号およびT2のすべてのデータを送信する
データの編集設定	無効
トラックのプレフィックス	なし
トラックのサフィックス	なし

9. 付録B：磁気ストライプの標準フォーマット

ISOクレジットカードフォーマット

ISOはInternational Standards Organizationの略です。

トラック1

フィールドID	内容	長さ
a	開始符号	1
b	フォーマットコード「B」	1
c	アカウント番号	12または19
d	セパレーター「^」	1
e	カード保有者の名前	可変
f	セパレーター「^」	1
g	有効期限4	
h	自由に使用できるオプションのデータ	可変
i	終了符号	1
j	水平冗長検査（LRC）文字	1

トラック2

フィールドID文字	内容	長さ
a	開始符号	1
b	アカウント番号	12または19
c	セパレーター「=」	1
d	有効期限「YYMM」	4
e	自由に使用できるオプションのデータ	可変
f	終了符号	1
g	水平冗長検査（LRC）文字	1

AAMVA運転免許証フォーマット

トラック1

フィールドID文字	内容	長さ
a	開始符号	1
b	都道府県	2
c	市町村	13
d	氏名	35
e	住所	29
f	終了符号	1
g	水平冗長検査（LRC）文字	1

トラック2

フィールドID文字	内容	長さ
a	開始符号	1
b	ANSIユーザー コード	1
c	ANSIユーザー ID	5
d	管轄ID/DL	14
e	有効期限	4
f	生年月日	8
g	残りの管轄	
i	ID/DL	5
h	終了符号	1
i	水平冗長検査（LRC）文字	1

トラック3

フィールドID文字	内容	長さ
a	開始符号	1
b	テンプレート バージョン番号	1
c	セキュリティ バージョン番号	1
d	郵便コード	11
e	クラス	2
f	制限	10
g	承認	4
h	性別	1
i	身長	3
j	体重	3
k	髪の色	3
l	目の色	3
m	ID番号	10
n	予約スペース	16
o	誤り訂正	6
p	セキュリティ	5
q	終了符号	1
r	水平冗長検査（LRC）文字	1

10. 付録C：その他のモードでのカード データの出力

特定のソフトウェア要件に適合できるようにするオプションのデータ出力フォーマットがMSRでサポートされています。

<01h> <01h> <1Ah> <02h> <00h> <左側の8バイトのデバイス シリアル番号> <6バイトのランダム データ>

<30h> <31h> <264バイトのサンプリング データ>。

11. 付録Dデータの暗号化および復号化の指針

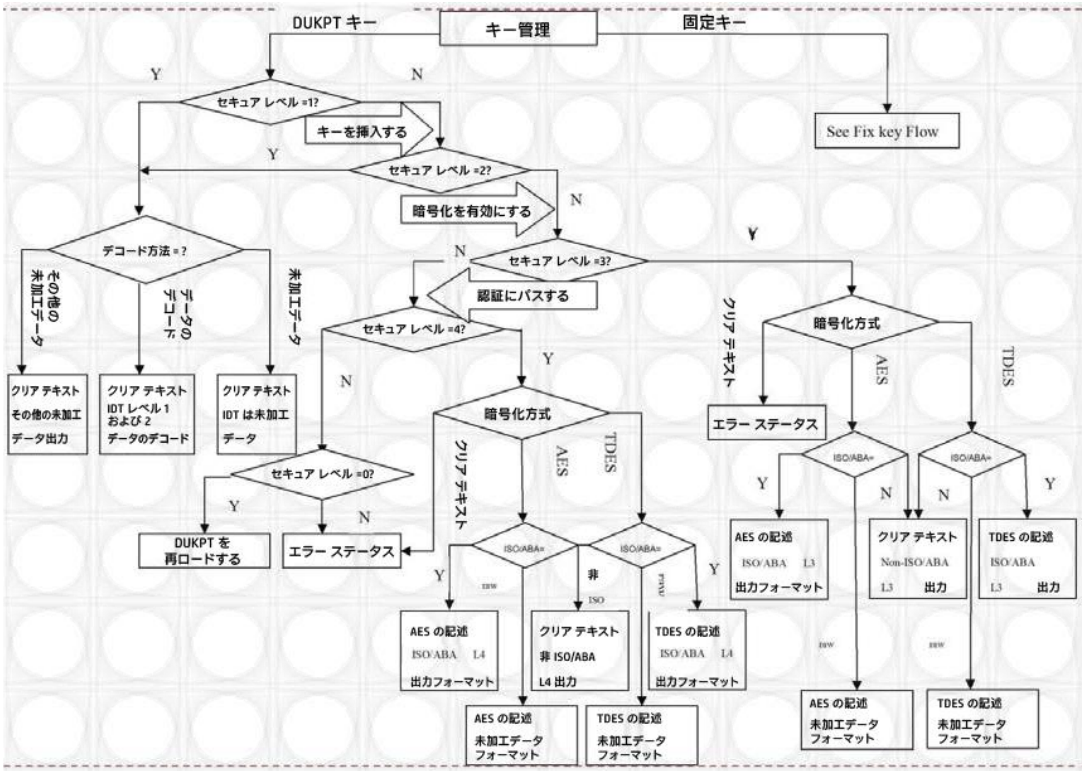
MSRで使用される暗号化モードは、暗号ブロック チェーン（CBC）と呼ばれます。この方式では、データの各ブロックが、前のデータ ブロックとXORされてから暗号化されます。各ブロックの暗号化は、前のすべてのブロックに依存します。そのため、暗号化された各データ ブロックを順番に復号化する必要があります。

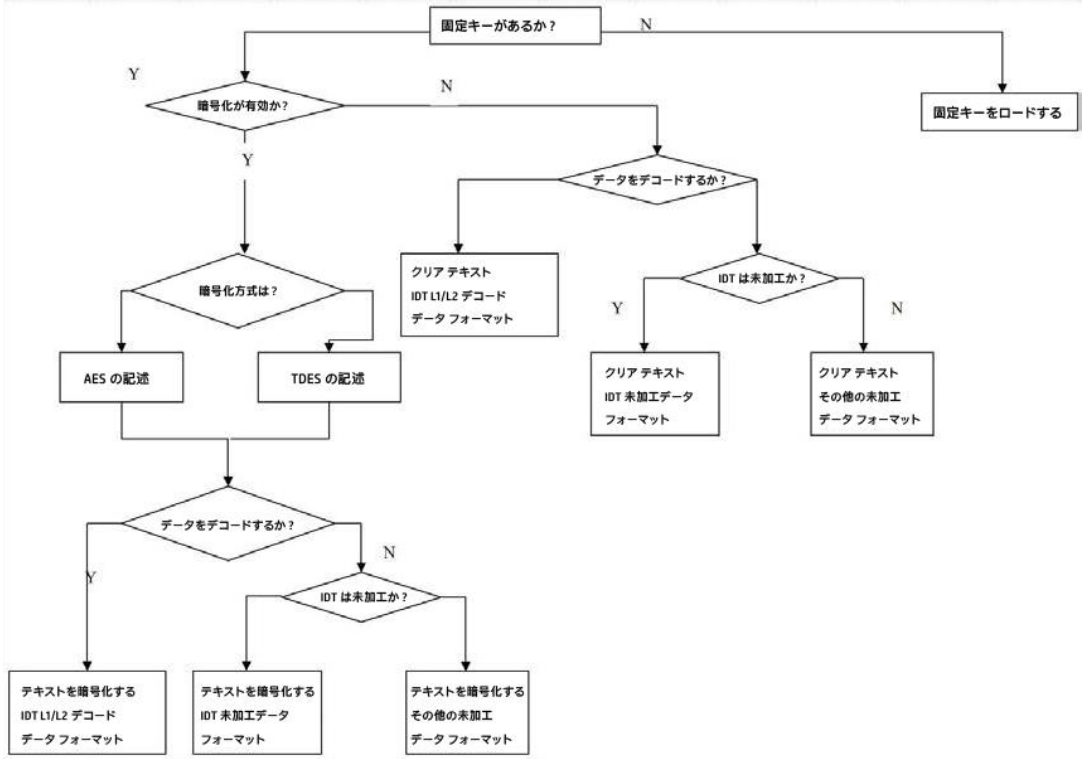
データを暗号化するには、最初に8バイトの0x00を生成して、暗号化前の最初のデータ ブロックとXORされる初期化ベクトルとして使用します。その後、データはTDESアルゴリズムを使用してデバイス キーで暗号化されます。結果は、暗号化前の次の8バイトのデータ ブロックと再度XORされます。すべてのデータ ブロックが暗号化されるまでこのプロセスが繰り返されます。

ホストでは、データを受信すると、ブロックの先頭から暗号テキストを復号化できます。ただし、暗号化テキストおよびクリア テキストの両方のデータを記録しておく必要があります。一方で、データを最後のデータ ブロックから最初のデータ ブロックへと逆方向に復号化することもできます。これにより、復号化が進行中であるときに、元のデータを復号化データに置き換えることができます。

リバース方式を使用してデータを復号化するには、最初にTDESの復号化を使用して最後の8バイトのデータを復号化します。次に、結果と前のデータ ブロックのXOR演算を実行して、最後のデータ ブロックをクリア テキストで取得します。最初のブロックに到達するまで、同じ方法でその次の前のブロックの復号化を続けます。最初のデータ ブロックでは、00hバイトとのXORを実行することになるため、XOR演算をスキップできます。

12. 付録E：キー管理のフロー チャート





13. 付録F：デコードされたデータの復号化の例

すべての例に使用されるキーは、0123456789ABCDEFFEDCBA9876543210です。

セキュリティ レベル3の復号化：拡張暗号化フォーマット

拡張暗号化フォーマットを使用した3トラックABAカードの復号化の例。MSRは、拡張暗号化構造フォーマットを除き、初期設定で設定されます。

拡張暗号化フォーマット（これは、下線が引かれた4番目のバイト（80）の上位ビットが1であるために認識できます。

```
029801803F48236B03BF252A343236362A2A2A2A2A2A2A393939395E42555348
204A

522F47454F52474520572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
2A2A

2A2A2A2A2A2A2A2A2A2A3F2A3B343236362A2A2A2A2A2A2A393939393D2A2A2A
2A2A

2A2A2A2A2A2A2A2A2A2A3F2ADA7F2A52BD3F6DD8B96C50FC39C7E6AF22F06ED1F0
33BE0
FB23D6BD33DC5A1F808512F7AE18D47A60CC3F4559B1B093563BE7E07459072ABF
8FAAB
5338C6CC8815FF87797AE3A7BEAB3B10A3FBC230FBFB941FAC9E82649981AE79F2
63215

6E775A06AEDAFAF6F0A184318C5209E55AD44A9CCF6A78AC240F791B63284E15B4
0191

02BA6C505814B585816CA3C2D2F42A99B1B9773EF1B116E005B7CD8681860D174E
6AD3

16A0ECDBC687115FC89360AEE7E430140A7B791589CCAADB6D6872B78433C3A25D
A9DD
AE83F12FEFAB530CE405B701131D2FBAAD970248A456000933418AC88F65E1DB7E
D4D10
973F99DFC8463FF6DF113B6226C4898A9D355057ECAF11A5598F02CA31688861C1
57C1C E2E0F72CE0F3BB598A614EAABB16299490119000000000206E203

STX, Length(LSB, MSB), card type, track status, length Track1,
length Track2, length Track3 02 9801 80 3F 48-23-6B 03BF
```

上記を分解して解釈しました02：

STX文字

98：全長の下位バイト 01：全長の上位バイト

80 : カード タイプ バイト（解釈では新しいフォーマットのABA カード）3F : 3トラックのデータがすべて良好

48：トラック1の長さ 23：トラック2の長さ 6B：トラック3の長さ

03：トラック1および2にマスクまたはクリア データあり BF：ビット7=1：KSNが含まれている

ビット6=0：セッションIDが含まれていないため、レベル4の暗号化ではない ビット5=1：トラック3のハッシュ データあり

ビット4=1：トラック2のハッシュ データあり ビット3=1：トラック1のハッシュ データあり

ビット2=1：トラック3の暗号化データあり

ビット1=1：トラック2の暗号化データあり ビット0=1：トラック1の暗号化データあり

マスクされたトラック1のデータ（長さ0x48）

```
252A343236362A2A2A2A2A2A2A393939395E42555348204A522F47454F524745
2057

2E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
2A2A

3F2A
```

ASCIIでのトラック1のマスク データ

```
%*4266********9999^BUSH JR/GEORGE
W.MR^*******************************?*
```

マスクされた16進数でのトラック2のデータ（長さ0x23）

```
3B343236362A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A
3F2A
```

ASCIIでのトラック2のマスク データ

```
;4266********9999=***************?*
```

この例では、トラック3のデータはクリア データでもマスク データでもありません（暗号化およびハッシュ データは次のとおりです）。トラック1の暗号化されたデータの長さ0x48は8バイトに切り上げられる= 0x48（10進数で72）。

```
DA7F2A52BD3F6DD8B96C50FC39C7E6AF22F06ED1F033BE0FB23D6BD33DC5A1F8
```

08512F7AE18D47A60CC3F4559B1B093563BE7E07459072ABF8FAAB5338C6CC88

15FF87797AE3A7BE

トラック2の暗号化されたデータの長さ0x32は8バイトに切り上げられる= 0x38 (10進数で56)。

AB3B10A3FBC230FBFB941FAC9E82649981AE79F2632156E775A06AEDAFAF6F0A
184318C5209E55AD

トラック3の暗号化されたデータの長さ0x6Bは8バイトに切り上げられる= 0x70 (10進数で64)。

44A9CCF6A78AC240F791B63284E15B4019102BA6C505814B585816CA3C2D2F42
A99B1B9773EF1B116E005B7CD8681860D174E6AD316A0ECDBC687115FC89360A
EE7E430140A7B791589CCAADB6D6872B78433C3A25DA9DDAE83F12FEFAB530CE
405B701131D2FBAAD970248A45600093

トラック1のハッシュされたデータ（長さ20バイト）:
3418AC88F65E1DB7ED4D10973F99DFC8463FF6DF

トラック2のハッシュされたデータ（長さ20バイト）:
113B6226C4898A9D355057ECAF11A5598F02CA31

トラック3のハッシュされたデータ（長さ20バイト）:
688861C157C1CE2E0F72CE0F3BB598A614EAABB1

KSN（長さ10バイト）: 62994901190000000002

LCR、チェックサム、およびETX 06E203 ASCIIでのクリアまたはマスク データ：

トラック1： %*4266********9999^BUSH JR/GEORGE

W.MR^********************************?* トラック2：

;4266********9999=***************?*

キーの値： 1A 99 4C 3E 09 D9 AC EF 3E A9 BD 43 81 EF A3 34 KSN: 62 99 49

01 19 00 00 00 00 02

復号化されたデータ：トラック1が復号化されました

%B4266841088889999^BUSH JR/GEORGE
W.MR^080910110000110000000046000000?!

トラック2が復号化されました

;4266841088889999=0809101100000046?0

トラック3が復号化されました

;333333333376767607070776767633333333337676760707077676763333333333767

6760707077676763333333333767676070?2

埋め込みのゼロを含む、16進数でのトラック1の復号化データ（ただし、ここには埋め込みのバイトはありません）:

254234323636383431303838383839393395E42555348204A522F47454F5247452057

2E4D525E3038303931303131303030303131303030303030303030343630303030303030

3F21

埋め込みのゼロを含む、16進数でのトラック2の復号化データ

3B343236363834313038383838393939393D303830393130313130303030303436 3F30

0000000000

埋め込みのゼロを含む、16進数でのトラック3の復号化データ

3B333333333333333333333736373637363037303730373736373637363333333333333

333333337363736373630373037303737363736373633333333333333333333337363736373630373037

3637363037303730373736373637363333333333333333333333736373637363037303037

3F320000000000

14. 付録G：HPの未加工データの復号化の例

元の未加工データ（順方向）

```
01D67C81020408102D448102040810204289OA350854A2FB3EE4BA3D4065B67A9C
391F

582A42B9

9A858A90AF60852B14AA628A0D

028FC210842C18421084030092040B51581F24B56074404811160D
```

元の未加工データ（逆方向）

```
01A28CAA51A9420DEA12A342B33A84A835F13872BCDB4C0578BA4EF9BE8A542158
A122

840810204081024568102040810 27CD60D02D11024045C0D5A49F03515A0409201
8042

10843068421087E20D
```

注：

- 各トラックの前にトラック番号があります。トラック1は01、トラック2は02、トラック3は03です。
- 各トラックの後にトラック セパレーターがあります：0D

元の暗号化フォーマットを使用した2トラックABAカードの復号化の例

この復号化は、固定およびDUKPTの両方のキー管理用です。

MSRには初期設定があります。

すべての例に使用されるキーは、0123456789ABCDEFFEDCBA9876543210です

元の暗号化フォーマット

元の暗号化フォーマット（これは、下線が引かれた4番目のバイトの上位ビットのために認識できます）：

(1)は0です。

```
028700041B331A0027D2E435CEE303F007E977B598B7E3C57C76F4445E309F6916
C032

1A0F915B6E490813498839049FE5204762327C3C758C5BF82542DEEDD8D6AF8801
9149
```

A702FF2D43BD4AD60031FA450720B00D7808E15F3D5B29AE712C64A1212E9AF6F4
83BD
40798A9FF2DDE77D046620B55BCE94A4D5534CF57E7E07629949011A0000000001
871D

03

STX, Length (LSB, MSB), card type, track status, length Track1, length Track2, length Track3 02 8700 04 1B 33 1A 00

トラック1および2の暗号化された長さ0x33+0x1Aは8バイトに切り上げる= 0x4D -> 0x50（10進数で80）

27D2E435CEE303F007E977B598B7E3C57C76F4445E309F6916C0321A0F915B6E49
0813

498839049

FE5204762327C3C758C5BF82542DEEDD8D6AF88019149A702FF2D43BD4AD60031F
A450 720B00 D7808

トラック1のハッシュ： E15F3D5B29AE712C64A1212E9AF6F483BD40798A

Track2 hashed: 9FF2DDE77D046620B55BCE94A4D5534CF57E7E07

KSN 629949011A0000000001

LRC、チェックサム、およびETX 87 1D 03

キーの値： 8A 60 A3 EB 80 87 63 52 B8 F5 05 CD A8 3C 33 70

KSN： 62 99 49 01 1A 00 00 00 00 01

復号化された未加工データ：

01D67C81020408102D448102040810204289OA350854A2FB3EE4BA3D4065B67A9C
391F

582A42B99A858A90AF60852B14AA628A028FC210842C184210840300920408515
1F24 B5607440481116

-=-
=-=-=-=-=-=-=-=-=-=-=-

=-=-=-=-=-

15. 付録H：SPI送信側チップの制御の例

```
/*H**************************************************************************
  * NAME:      spi_drv.h
  * Copyright (c) 2003 ID TECH.
  * RELEASE:   cc03-demo-spi-0_0_1
* REVISION:    1.1.1.1
  * PURPOSE:
  * spi lib header file

**************************************************************************/


#ifndef
_spi_DRV
_H_
#define
_spi_DRV
_H_

/*______I N C L U D E S_____*/

/*_____D E F I N I T I O N__*/
// Pin define
 #define _DAV_IN                P3_4                                    // SPI
 chip has data ready
 #define _SPI_SS                P1_1                                    // SPI
 chip select pin

//In Sender mode, the baud rate can be selected from a baud rate generator which is controlled
//by three bits in the SPCON register: SPR2, SPR1 and SPR0. The Sender clock is
//chosen from one of seven clock rates resulting from the division of the internal clock by
//2, 4, 8, 16, 32, 64 or 128.

#define SPI_RATIO_2   0x00 // FCLK PERIPH/2
#define SPI_RATIO_4   0x01 // FCLK PERIPH/4
#define SPI_RATIO_8           0x02 // FCLK
PERIPH/8 #define SPI_RATIO_16 0x03 // FCLK
PERIPH/16
#define SPI_RATIO_32   0x80 // FCLK PERIPH/32
#define SPI_RATIO_64  0x81 // FCLK
PERIPH/64 #define SPI_RATIO_128       0x82
// FCLK PERIPH/128 #define
SPI_RATIO_INVALID      0x83 // No
BRG

/*     M A C R O S     */
// SPIF: Serial Peripheral data transfer flag
// Cleared by hardware to indicate data transfer is in progress or has been
// approved by a clearing sequence.
// Set by hardware to indicate that the data transfer has been completed.
#define Spif_set()    ((SPSCR & MSK_SPSCR_SPIF) == MSK_SPSCR_SPIF) // If equal, the
data transfer has been completed.
/*_____D E C L A R A T I O N */ Uchar spi_set_speed(Uchar data
ratio);
void spi_sender_init(Uchar data cpol, Uchar data cpha, Uchar data ssdis, Uchar
data speed); void spi_Sendout(Uchar data inchar);

#endif /* _SPI_DRV_H_ */
/*C**************************************************************************
  * Module:    main.c
/**************************************************************************
  * Copyright (c) 2004 ID TECH inc.,
/**************************************************************************
  * CREATION_DATE:     2004.1.10
/**************************************************************************
  * PURPOSE:
```

```
  * spi library low level functions (init, receive and send functions)
  * and global variables declarations to use with user software application
*****************************************************************************/
/*_____I N C L U D E S_____*/ #include "spi_drv.h"
/*_____M A C R O S_____*/ #define MAX_LEN    512
/*_____D E F I N I T I O N___*/

Uchar data SPI_IPNT; // Temp buffer to store SPI data.
Uchar data Command_OUTbuf[MAX_LEN]; // Command
output buffer Uchar data Command_INbuf[MAX_LEN];
// Command input buffer Uint16 data spilength; //
received command length
Uint16 data Command_Length; // output command length
/*_____D E C L A R A T I O N */

void main(void){
Uint16 data i, j; // Internal counter.

        spi_sender_init(0, 0, 1, 32); //SPI sender mode, initialize to CPOL=0,
CPHA=0, SSDIS=1, bitrate=Fper/32
Enable_spi_interrupt(); // Turn on SPI interrupt in system.
_SPI_SS = 0; // Disable SPI receiver during power on, to prevent indeterminate state.

do{ // keep polling...
{
// .............       Other subroutine to handle other tasks
}

if(_DAV_IN){ // If DAV pin is high level, SPI receiver has data ready.
                      _SPI_SS = 1; // To Generate a falling edge. Not useful for clock phase
0, but clock phase 1 needs this falling edge.
delay10us(); // Wait for high level get steady.
_SPI_SS = 0; // Pull chip select pin low, ready to start SPI communication. spilength =
0; // Initialize Command_buf pointer.

                    while(_DAV_IN){ // Keep polling DAV pin till it turns low level.
Polling interval is 40us in this demo code.

in this subroutine too.


buffer.
spi_Sendout(0xff); // Send out any data to get SPI receiver input, delay 40us
Command_INbuf[spilength++] = SPI_IPNT; // Save data into Command_buf. if(spilength >= MAX_LEN){
// Quit while loop if read the end of input
break;
}


high.
```

```
}
_SPI_SS = 1; // Read out all the data from SPI receiver, set chip select pin to idle

for(i = 0; i < spilength; i++){ // Send out data from UART port.
put_byte(Command_INbuf[i]);
}
}

{
// .............      Other subroutine to handle other tasks
}

if(SPISenderCommandReady){ // If SPI sender wants to send a command to SPI receiver
                        _SPI_SS = 1; // To Generate a falling edge. Not useful for clock
phase 0, but clock phase 1 needs this falling edge.
delay10us(); // Wait for high level get steady.
_SPI_SS = 0; // Pull chip select pin low, ready to start SPI communication.

for(j = 0; j < Command_Length; j++){ // Send out whole command string.
spi_Sendout(Command_OUTbuf[j]);


chip select pin to idle high.

}

{

tasks
}
```

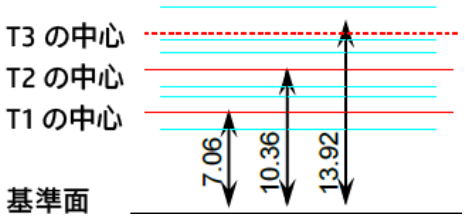
```
}

       _SPI_SS = 1; // Read out all the data from SPI receiver, set BeepOn_Long(); // Send
out one beep to indicate command finished.

// .............      Other subroutine to handle other
}
while( TRUE );
}
/*C**************************************************************************
  * Module:    spi_drv.c
/**************************************************************************
  * Copyright (c) 2004 ID TECH inc.,
/**************************************************************************
  * CREATION_DATE:     2004.1.10
/**************************************************************************
  * PURPOSE:
  * spi library low level functions (init, receive and send functions)
  * and global variables declarations to use with user software application
**************************************************************************/
/*______I N C L U D E S_______*/ #include "spi_drv.h"
/*______M A C R O S___*/
/*______D E F I N I T I O N___*/ Uchar transmit_completed = 0; //
0 by default
extern Uchar data SPI_IPNT;
/*______D E C L A R A T I O N */

// Here are some global flags to use with spi library
// These global flags are used to communicate with higher level functions ( user application )
// Here the global variables to communicate with spi interrupt routine

/*F**************************************************************************
  * NAME: spi_isp
*
  * PARAMS: none
  * return: none
  * PURPOSE:
  * spi - interruption program for serial transmission ( Sender and Receiver mode )
*
  * NOTE:
**********************************************************************
**********/ Interrupt(void spi_isp(void), IRQ_SPI){
if(Spif_set()){ // Quit if data transfer has not been
completed. transmit_completed = 1;   // Set software complete
flag
SPI_IPNT = SPDAT;      // Store SPI input data in SPI_IPNT. SPDAT - Serial Peripheral
Data R
e
g
i
s
t
e
r
return
  ;
   }
}
/*F**************************************************************************
  * NAME: spi_set_speed
  * PARAMS: ratio: spi clock ratio/XTAL
  * return: Uchar: status
  * PURPOSE:
  * Configure the baud rate of the spi, set CR2, CR1, CR0
  * NOTE:
  * This function is only used in spi sender mode, called by spi_sender_init
**********************************************************************
**********/ Uchar spi_set_speed(Uchar data ratio){
switch(ratio){ // Set SPCON register
case 2: SPCON |= SPI_RATIO_2; break; // FCLK PERIPH/2 case 4: SPCON
```

```
|=  SPI_RATIO_4;  break;  //  FCLK  PERIPH/4  case  8:  SPCON  |=
SPI_RATIO_8; break; // FCLK PERIPH/8
case 16:SPCON |= SPI_RATIO_16;break; // FCLK PERIPH/16
case 32:SPCON |= SPI_RATIO_32;break; // FCLK PERIPH/32
case 64:SPCON |= SPI_RATIO_64;             break; // FCLK
PERIPH/64 case 128:SPCON |= SPI_RATIO_128; break; // FCLK
PERIPH/128
default :      return FALSE;
}
return TRUE;
}
/*F**************************************************************************
  * NAME: spi_sender_init
  * PARAMS:
  * cpol: Uchar CPOL value
  * cpha: Uchar CPHA value
  * ssdis: Uchar SSDIS value
  * speed: Uchar spi speed ratio transmission Vs Fper
  * return:    none
*
  * PURPOSE:
  * Initialize the spi module in sender mode
*
  * EXAMPLE:
  * spi_sender_init(0,0,1,32); // init spi in mater mode with CPOL=0, CPHA=0,
  * // SSDIS=1 and bitrate=Fper/32
  * NOTE:
**************************************************************************/
void spi_sender_init(Uchar data cpol, Uchar data cpha, Uchar data ssdis, Uchar
data speed){ SPCON = 0; // Initialize SPCON: Serial Peripheral Control Register
SPCON |= MSK_SPCON_MSTR; // Serial Peripheral Sender: Set to configure the SPI as a Sender.
_SPI_SS = 1; // Initialize chip select pin to idle - high
level. spi_set_speed(speed); // Set SPI sender speed to
Fper/32.
if(cpol) SPCON |= MSK_SPCON_CPOL; // Cleared to have the SCK set to “0” in idle state.
if(cpha) SPCON |= MSK_SPCON_CPHA; // Cleared to have the data sampled when the SCK leaves
the idle
state
if(ssdis)      SPCON |= MSK_SPCON_SSDIS; // Set to disable chip select in both Sender and
Receiver modes. Select manually control CS pin.
SPCON |= MSK_SPCON_SPEN; // Set to enable the SPI interface.
}
/*F**************************************************************************
  * NAME: spi_Sendout
  * PARAMS: inchar: the desired character to send out
  * return: none
*
  * PURPOSE:
  * Send out one character
  * NOTE:
  * This function is use only in spi sender mode
*****************************************************************
**********/ void spi_Sendout(Uchar data inchar){
Uchar data m;
SPDAT = inchar;// send a data, put the data into SPDAT register
while(!transmit_completed);// wait for transmission complete (interrupt
complete), flag
transmit_completed will be set in SPI interrupt
subroutine. transmit_completed = 0;   // clear software
transmit end flag
m = 4; // Delay 40us then poll for DAV pin status or send out next
byte. do{
delay10us();
}while(m--)
```

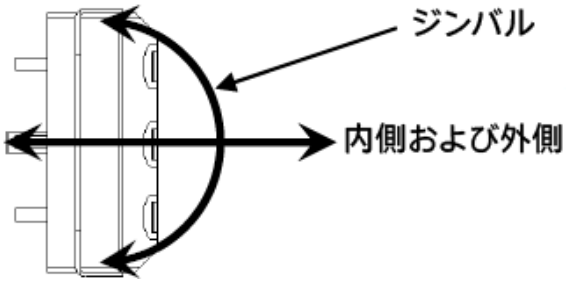
16. 付録I：磁気ヘッドの機械的設計のガイドライン

この取り付けガイドは、スプリング マウント付きのHPの磁気ヘッドを取り付けるときに特に使用します。

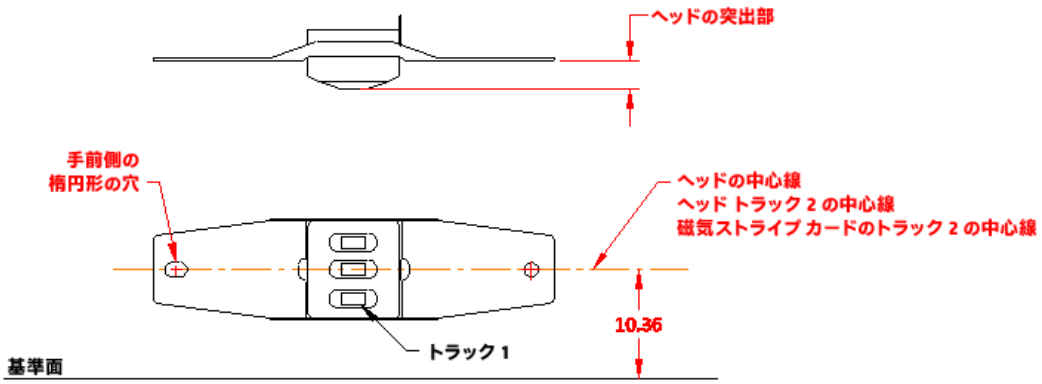
1. ISO 7810およびISO 7811規格には、「標準」のすべての磁気ストライプ カード向けの仕様が定義されています。各磁気トラックの中心線の適切な位置を以下の図に示します（**注：**カードの基準面はカードの端です。そして、磁気ヘッドに言及する場合、それはカードを乗せる面になります）。



2. ヘッドの取り付けにより、ヘッドはカードの磁気ストライプに追従できるようになります。つまり、磁気ヘッドは、レールに取り付けられた後、カードの表面と接触し続けるためにジンバル（トラック2の中心線の周りを回転すること）や移動が自由にできる必要があります。次の図は、ヘッドの取り付けで可能にする必要のある回転運動および直線運動を示しています。



3. ヘッドの磁気トラックが、カード上の磁気トラックと同じ基準（スロットの底部）からの距離に位置するように、カードがスライドされる基準面に対してヘッドを取り付ける必要があります（上記の1番を参照してください）。次の図に、スプリング付きの一般的なHPの磁気ヘッドを示します。スプリングの取り付け穴（トラック2の中心線を中心とする）は、ヘッドの取り付けとトラックの場所の位置決めに使用されます。(**注：**トラック1～3の場所を適切に決めるために、スプリングの楕円形の穴を図面に示されている向きにする必要があります。



ヘッドの中心線が基準面と平行になるようにしてください。

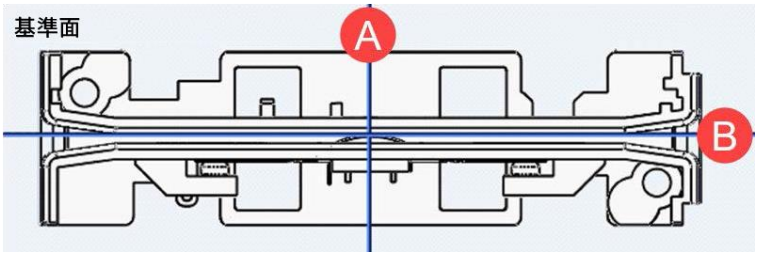
4. レールおよびヘッドの取り付けを設計するときは、カードの厚さを考慮する必要があります。(ヘッドの頂部にある) ヘッドと反対側のカード スロットの壁の間の距離は、最小の厚さのカードがスワイプされたときに少なくとも0.25 mmの動きがあるように調整する必要があります。動きが少ないと、読み取りの信頼性が低下する可能性があります。言い換えると、ヘッドの頂部と反対側のスロット壁の間の距離は、リーダーを通る最小のカードの厚さのほんのわずかにすぎないため、磁気ヘッドは常にカードに圧力を加えます。この圧力により、特に高速でのヘッドからストライプへの適切な接触が可能になります。

5. 標準のカードの厚さは0.76 mm ± 10%です。標準のカードのみを使用する場合は、ヘッドの頂点 (ヘッドの頂部) を反対側のカード スロット壁から最大0.25 mm離すようにしてください。標準よりも薄いまたは厚いカードを使用する場合は、反対側の壁からのヘッドが配置される距離を調整する必要があります (これには、カードのスロット幅が広いか狭い一意のレール デザインが必要になります)。

6. 最小スロット幅は、カードの最大の厚さに0.15 ~ 0.30 mmを加えたものにしてください。標準のカードを使用する場合の推奨される最小スロット幅は1.03+0.08 -0 mmです。

7. このデザインでは、ヘッド スプリングの永久歪みを防ぐために、レールへのヘッドの取り付け中またはヘッドの取り付け後に、ヘッド スプリングの過剰な力や歪みが生じないようにする必要があります。ヘッド スプリングは、スプリング ホールの周りを自由に回転できるように取り付ける必要があります。

8. スロットの底部およびスロットの壁に切れ目があってはならず、平らである必要があります（歪みは許されません）。磁気ヘッドの頂部の両側にある約10 mmのスロット壁の部分は、傾斜があってはならず、スロットの底部（基準面）に垂直である必要があります。導入および導出部分のスロット幅はより大きくし、カードのスムーズなスワイプを妨げるエッジや角度を付けずに徐々に推移していく必要があります。



A. ヘッド ギャップ面

B. スロット壁面

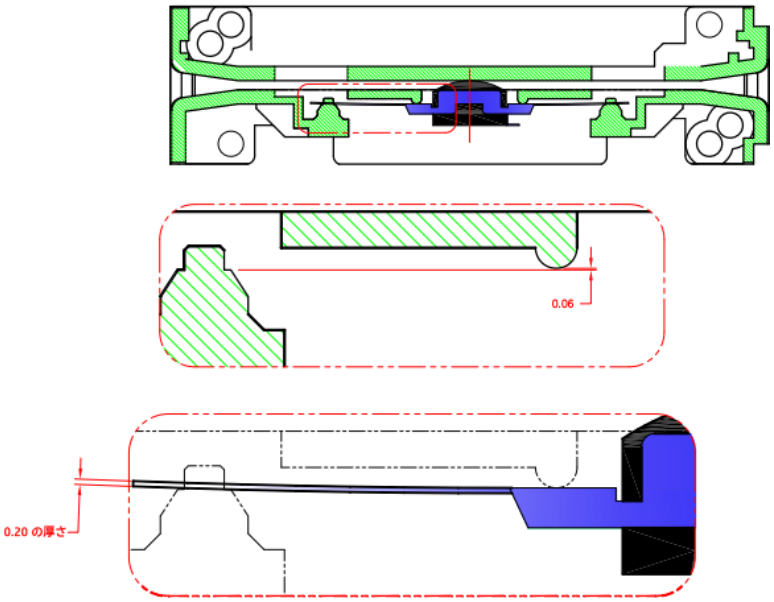
9. スワイプのライフサイクルの要件に応じて、レールに適した素材を決める必要があります。リーダーの寿命が50,000回のパスを超える場合は、スロットの底部に金属製の摩耗プレートを埋め込む必要があります（ステンレスは腐食を防ぐために望ましい金属です）。または、スロットに使用されるプラスチック素材をカードの素材よりもかなり硬くして、レールの寿命を十分に確保する必要があります。

10. 磁気ヘッドの背面（ピン側）には、最大の厚さのカードをスワイプしているときに他の部品との干渉が発生するのを防ぐための十分な予備空間が必要です。このデザインでは、カードのスワイプ中に適切な回転やヘッドの動きを可能にするために、ヘッドの後ろに最低1.25 ~ 1.52 mmの空間を設ける必要があります。レール内のヘッド開口部には、最大の回転アクションを行えるだけの空間を持たせる必要があります。

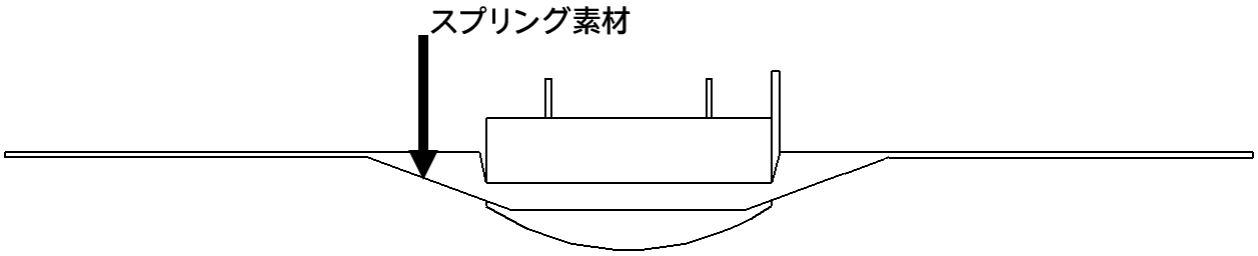
11. ヘッドをレール内に取り付ける場合は、ヘッドを保持するスプリングを少しだけプリロードしてください。スプリングをプリロードしておくと、ヘッドがカードによる衝撃を最初に受けたときに、ある程度の安定性を確保できます。これは、カードを高速でスワイプする場合に特に重要です。（スプリングがプリロードされていない場合、カードによる衝撃を受けたときにヘッドが振動する傾向があり、この振動によってヘッドがカードとの接触を失います。）

12. ヘッド スプリングをプリロードするHPのソリューションでは、レール内に成形された2つの左右対称の突起をヘッド（ヘッド窓部）の両側に1つずつ追加します（下の図面を参照してください）。スプリングの静止面と突起の頂部との差が0.06+/-0.03 mmになることをおすすめします。これにより、0.20 mmの厚さのヘッド スプリングの場合、0.14+/-0.03 mmのしなりが生じます。

突起は円筒形で、その頂部がヘッドの頂部の反対側のスロット壁と平行になるようにしてください。これにより、ヘッドがレールに取り付けられたときに、その頂部がスロット面と平行になり、カードの磁気ストライプと適切に接触するようになります。下の図面を参照してください。



13. レールのスロットの長さ、幅、および高さは、読み取り性能の安定性に影響します。
    a. スロットの長さは、寸法の制約によって許容される最大の値にします（可能であれば、カードの長さの2倍にしてください)。
    b. スロットの幅は、スロットに通す最大の厚さのカードよりも0.20 mmほど大きくします。
    c. スロットの高さは、寸法の制約で許される限り大きくしますが、カードのエンボス加工部分に及ばないようにしてください（ただし、レール壁面のデザインにそのようなエンボス加工を許可する対策（くぼみ）がある場合を除きます)。
14. ヘッドがスロット内に突出するときに通るレール壁面の窓部は、ヘッドが自由に動作できるだけの十分な大きさにしてください。
15. ヘッドとレール窓部の壁面の間の隙間は、ヘッドの移動量およびヘッドの突出部（ヘッドの頂部からスプリング表面までの距離）によって異なります。
16. 1.03 $^{+0.08}_{-0}$ mm幅のスロットがある標準のレールおよび3.50 mmのヘッド突出部があるID Techの標準のヘッドの場合、そのヘッドのすべての側面で最低1.25 mmの隙間が確保されている必要があります。(注：このガイドラインは、ロー プロファイルのヘッドが使用されている場合には適用されません。以下の図に示すように、この窓部では、磁気ヘッドに溶接されたスプリングの部分に隙間を確保する必要があります。)



HPでは、デザイン参照用のレールおよび磁気ヘッドのサンプルを提供できます。以下の製品番号を使用して、最寄りの営業担当者にご注文ください。

- 90 mmレール：80006248-001
- 標準ウィング スプリング ヘッド：80027236-001

17. 付録J：ファームウェアのアップグレード

HP TM4 SPIのMSRファームウェアは、SPI通信ポート経由で更新できます。

HPでは、Windowsベースのユーティリティ ソフトウェアであるFWUpdate.exe、および参照用にRS-232からSPIへのコンバーター ボードを提供できます。お客様は、ファームウェアをアップグレードするための独自のソフトウェアを開発することもできます。(前提条件：ホストがすでにMSRと通信している必要があります。「ファームウェア バージョンの読み取り」などの通常のコマンドがサポートされている必要があります。)

手順

TriMag IVファームウェアは、以下のコマンドを使用して更新できます。

特に明記されていない限り、コマンドはSTX（0x02）およびETX（0x03）でラップし、その後ろに1バイトのLRC（STXおよびETXを含む先行するすべてのバイトをXORで算出したもの）を続ける必要があります。

また、特に明記されていない限り、正常な応答はACK（0x06）で始まります。

基本的なステップ

1. ファームウェア バージョンを読み取ります（52 22 88コマンド）。これは、現在のリーダーが機能していることを確認するためです。
2. ファームウェアを消去します（53 7E 0D 31 01 02 03 04 05 06 07 08 04 03 02 01）。

ファームウェアが約2秒で消去されたら、DAV線を立ち上げて0x5Aの送信を要求します。ホストでこの応答を読み取る必要があります。

注：DAV線は500 mSで「ハイ」になっています。ソフトウェアが応答を読み取らない場合、MSRではRS232通信に移行します。このような場合、5Aバイトを取得するには、MSRの電源を入れ直し、500 mSのDAVが「ハイ」である期間内に応答を読み取る必要があります。

応答を読み取った後、さらに3秒間待ってから、以下のロード シーケンスを実行することをおすすめします。

新しいファームウェアのロード

1. 16進バイト0xBDを送信して、ロードを開始します。
2. ファームウェアのbinファイルを開き、ファイル全体をMSRに送信します。

注：新しいファームウェア ファイルは、26 KBの暗号化されたファームウェアおよび4バイトのチェックサムとLRCを含むバイナリ ファイルです。チェックサムとLRCは、MSRによってチェックされます。MSRでは、ファームウェアのダウンロードを拒否するか受け入れるかを決定します。(ホストではこれらのバイトをチェックする必要はなく、ファイル全体を送信するだけです。)

1. DAV線が「ハイ」になるのを待って、1バイトの応答を読み取ります。
2. 3秒間待ちます。

例

HP FWupdateソフトウェアを使用してファームウェアをロードする場合の例を以下に示します。

手順1：現在のファームウェア バージョンを確認する：

出力	02 52 22 88 03 f9		
入力	06 02 49 44 20 54 45	43	..ID TEC 250ms
	48 20 54 4d 34 20 53	65	H TM4 Se
	63 75 72 65 48 65 61	64	cureHead
	20 53 50 49 20 52 65	61	SPI Rea
	64 65 72 20 56 31 2e	32	der V1.2 4.049..

B. 現在のファームウェアの消去：

出力	02 53 7e 0d 31 01 02	03	.S..1... 18sc
	04 05 06 07 08 04 03	02	.....
	01 03 1c	...	
IN			22

注：MSRでファームウェアの消去を完了するには約2秒かかります。ホストではDAV線が立ち上がるのを待って、応答5Aを読み取る必要があります。ホストでは、さらに3秒間待って、以下のロード ステップを実行する場合があります。

手順2：ファームウェアをダウンロードする

1. ダウンロード モードに入るための1バイトを送信します：BD
2. 暗号化されたbinファイル（新しいファームウェア ファイル）を送信します。
3. DAV線が立ち上がるのを待って、1バイトの応答を取得し、それを無視します。
4. 数秒（約3秒間）待ちます。

手順3：新しいファームウェア バージョンを確認する

出力	02 52 22 88	03 f9	.R"...	5.0sc
入力	06 02 49 44	20 54 45 43	..ID TEC	251ms
	48 20 54 4d	34 20 53 65	H TM4 Se	
	63 75 72 65	48 65 61 64	cureHead	
	20 53 50 49	20 52 65 61	SPI Rea	
	64 65 72 20	56 31 2e 32	der V1.2	
	34 2e 30 35	30 03 1f	4.050..	