

BVCM-MIF-02-IM10

Application Manual

株式会社ピコシステムズ

千葉県松戸市幸谷 3 3 2-7

TEL : 03-4431-1034

FAX : 03-4431-1035

§ 1 概要

BVCM-MIF-02-IM10 は NEC 製電子決済端末 IM10 を使った自動販売システムを、最低限のコマンドで制御することを目指して開発されました。

電子式決済端末では、商品金額が決定してから決済処理を開始するため、商品先選択方式の自動販売機となります。そのため、BVCM-MIF-02-IM10 はコインメック、ビルバリのサポートは行わず、商品先選択方式の自動販売機に特化した電子マネー決済専用のボードとなっています。

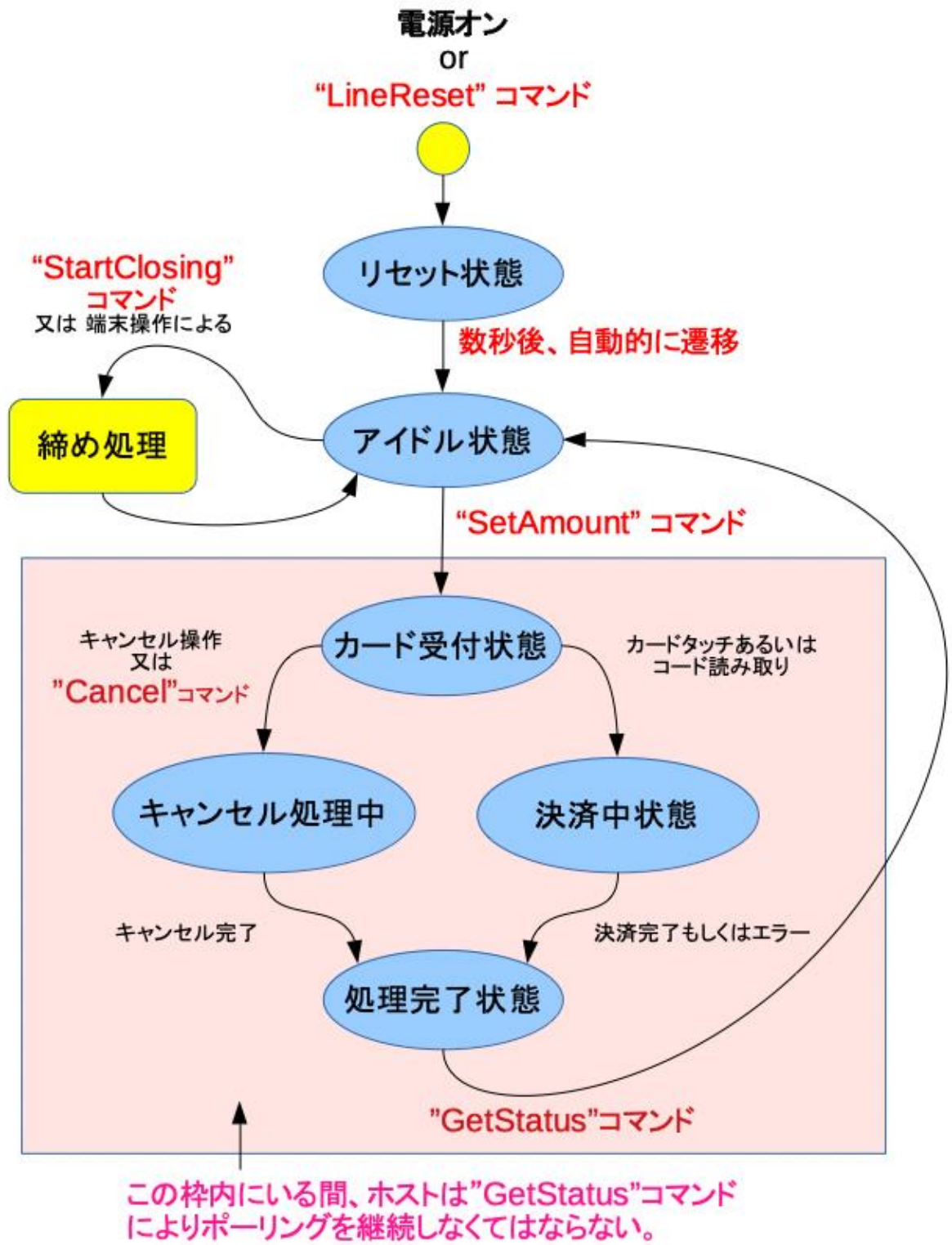
商品先選択方式

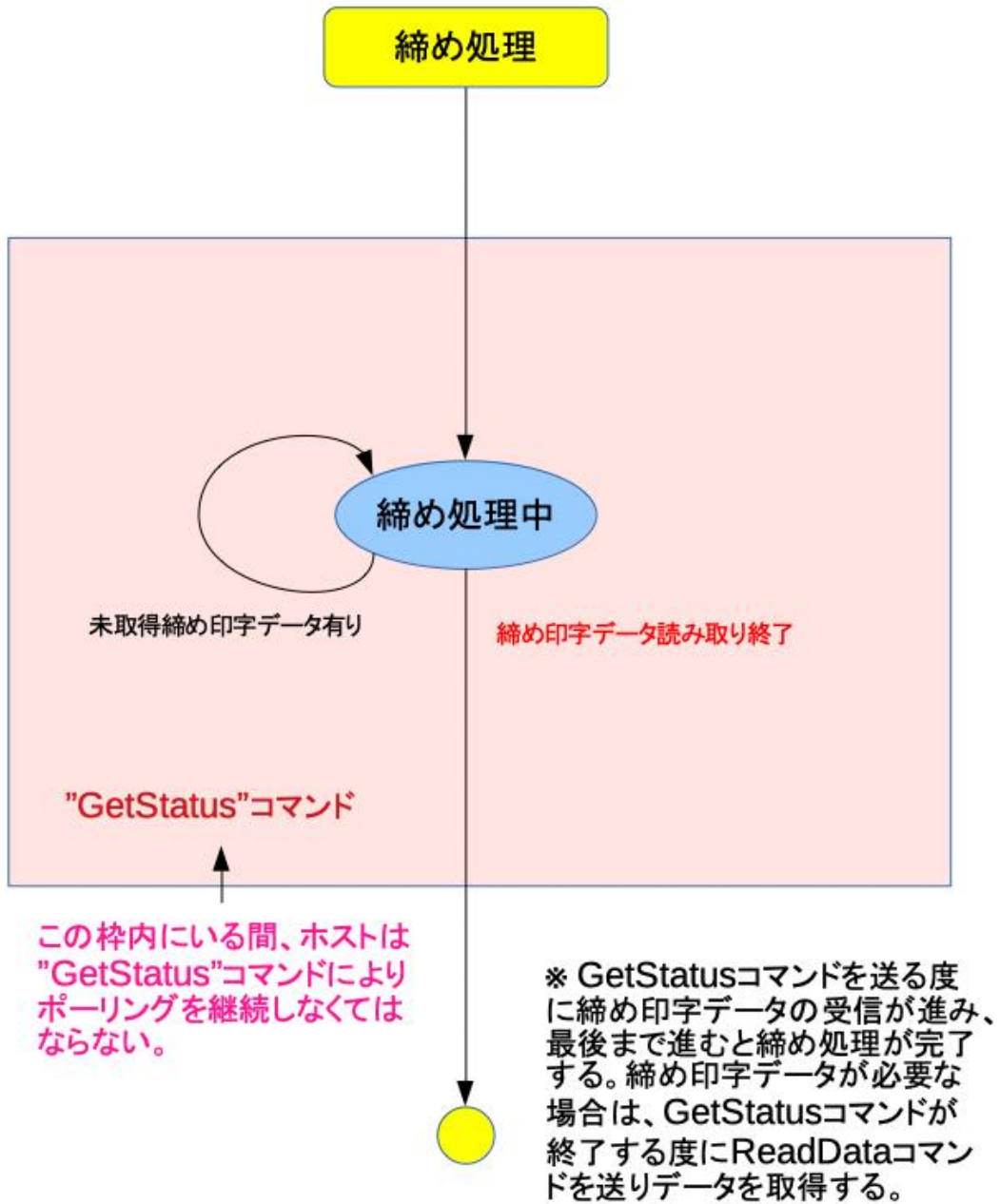
商品を先に決定し、金額が確定してから清算を開始します。自動販売機が BVCM-MIF-02-IM10 に商品と決済金額を設定すると、BVCM-MIF-02-IM10 は自動的に可能な決済方式により決済を行い、決済が完了すると決済完了ステータスを返します。自動販売機は決済完了ステータスを確認して商品を排出します。

§ 2 仕様および制限事項

- 対応電子マネー端末（NEC 製 IM10）
- レシート印刷対応
- 締め処理対応（コマンド起動および手動起動）
- 清算金額の最大値は 10 万円
- 決済金額を設定すると決済状態に移行し、カードのタッチあるいは読み込みで自動的に決済完了に移行する。
- 決済前の清算金額の変更をサポート。
- 決済せずに時間が経過するとタイムアウト
- 決済端末のキャンセルボタンをサポート（IM10 はキャンセルボタン無し）
- 決済動作中、ホストは 500msec. 以内に一度、情報取得コマンドによるポーリングを行わなければならない。本機はポーリングにより決済端末の状態を取得し内部情報を更新する事により動作（ポーリング駆動）する。ポーリングは処理が完了し、アイドル状態に戻るまで続ける必要がある。

§ 3 BVCN-MIF-02-IM10 状態遷移図





- **リセット状態**

電源投入あるいは LineReset コマンド発行後にこの状態になります。それ以外の原因でこの状態になった場合は、システムに何らかの異常が発生したと判断します。

- **アイドル状態**

リセット状態から数秒経過すると自動的にこの状態になります。あるいは以前の決済が完了した場合もこの状態に戻ります。BVCM-MIF-02-IM10 は何の処理も行わず、決済処理開始を待っている状態です。

- **カード受付状態**

SetAmount コマンドにより決済を開始します。一度、本状態に入ると BVCM-MIF-02-IM10 はポーリング駆動動作となり次にアイドル状態に戻るまで、ホストは GetStatus コマンドによるポーリングを続け続ける必要があります。

- **決済中状態**

決済端末の内部状態で、カードがタッチあるいは読み取られると、本状態に移行し、決済端末は決済システムのサーバーと通信し決済を行います。

- **キャンセル処理中**

カード受付状態で、キャンセルコマンドを受信した場合、本状態に移行します。ただし、すでに決済処理が走っている場合、キャンセルすることは出来ず、キャンセル不能ステータスが戻ります。

- **処理完了状態**

決済完了あるいはキャンセル完了すると処理完了状態となります。この状態をホストが確認 (GetStatus) するとアイドル状態に戻ります。

- **締め処理中**

StartClosing コマンドあるいは端末の操作により締め処理が開始されるとこの状態になります。この状態で GetStatus コマンドが実行されるとその都度、締め印字データが IM10 より読み出され、締め処理が進んでいきます。全ての印字データが読み出されるとアイドル状態に戻ります。

§ 4 コマンドおよびレスポンス

BVCM-MIF-02 Easy を制御するには、ホストからコマンドを発行し、それに対するレスポンスを BVCM-MIF-02 Easy から受け取るという手順で行われます。コマンド及びレスポンスの形式は以下の通りです。

コマンド形式

コマンド文字列は以下の形式（全 ASCII 文字）

コマンド + “,” + パラメータ + “,” {+ パラメータ + “,” }+ チェックサム + [CR]

コマンド : コマンド文字列

パラメータ : ASCII 文字列（コマンドによってパラメータの数は異なる。）

チェックサム : 最初の文字からパラメータ最後のカンマまで、ASCII 文字を 1 文字ずつ単純バイナリー加算し、下位 1 バイトを ASCII 文字 2 文字で表す。

※ 途中で空白は許されない。

レスポンス形式

レスポンス文字列は以下の形式（全 ASCII 文字）

レスポンス + “,” + データ + “,” {+ データ + “,” }+ チェックサム + [CR]

レスポンス : レスポンス文字列（OK, NG or BUSY etc.）

データ : ASCII 文字列（コマンドによってデータの数は異なる。）

チェックサム : 最初の文字からパラメータ最後のカンマまで、ASCII 文字を 1 文字ずつ単純バイナリー加算し、下位 1 バイトを ASCII 文字 2 文字で表す。

※コマンドエラー時にはレスポンスを返さない。

§ 5 コマンド詳細

- **情報取得コマンド (GetStatus) パラメータ数:0、応答データ数:6**

送信文字列 : GetStatus,D0[CR]

受信文字列(例): OK,02,,0,00,0,0,20[CR]
NG,C1[CR] (NG 時にはデータを返さない)

応答データ:

data1: 処理状態(16進値を ASCII 文字2文字で表す)

- 0x00: リセット起動中
- 0x01: アイドル状態
- 0x02: 決済中(カード受付中)
- 0x03: 決済完了状態
- 0x04: キャンセル中状態
- 0x05: 処理未了状態継続中
- 0x06: 締め処理中
- 0x07: 締めスイッチ検出(アイドル中)

data2: カードID(決済 OK 時: ASCII 文字列16文字)

data3: 決済後カード金額(決済 OK 時: ASCII 数値__ゼロサプレス)

data4: エラー情報(16進値を ASCII 文字2文字で表す)

- ビット 6: 電子マネー端末通信不能
- ビット 5: ビルバリ通信不能
- ビット 4: コインメック通信不能
- ビット 2: 電子マネー端末エラー
- ビット 1: ビルバリエラー
- ビット 0: コインメックエラー

data5: 決済結果(16進値を ASCII 文字2文字で表す)

- 00: 決済未確定
- 01: 決済 OK
- 02: 決済キャンセル
- 03: 決済 NG

data6: 金額表示(ASCII 数値__ゼロサプレス)

- 現金入金受付時 : 投入金額
- 決済 OK 時 : 決済金額
- 決済キャンセル時 : 0

※data1 で、締めスイッチ検出を受け取った場合、IM10 が連動締めで締めスイッチが操作されたことを示します。本状態を受け取った場合、IM10 単独で締め処理に入ることはなく、StartClosing コマンドを発行して締め操作を開始します。

- **金額設定コマンド (SetAmount) パラメータ数:2、応答データ数:6**

パラメータ:

param1: 清算金額(ゼロサプレス可)
param2: カラム番号(1~255)

例1) カラム1商品、100 円を設定する場合

送信文字列(例): SetAmount,100,1,E6[CR]

受信文字列: GetStatus コマンドに対する応答データと同様

- **キャンセルコマンド (Cancel) パラメータ数:0、応答データ数:6**

清算をキャンセルします。

送信文字列: Cancel,02[CR]

受信文字列: GetStatus コマンドに対する応答データと同様

- **リセットコマンド (LineReset) パラメータ数:6、応答データ数:0**

ハードウェアリセットおよび初期化を行います。

パラメータ:

param1: 自販機メーカーコード 0 - 255 (不明時は0を指定)
param2: 自販機種別 0 - 255 (不明時は0を指定)
param3: ソフトウェアバージョン 0 - 255 (不用時は0を指定)
param4: 現在時刻データ(15文字)
param5: 商品先選択タイマ値 0 - 255 [秒]
param6: レシート印字有効、無効(0:無効, 1:有効)

例2) 決済タイムアウト時間、20秒、2021年9月26日(日)13時25分00秒の場合

注: 曜日は日~土を0から6で指定(日付の次の一桁)

送信文字列(例): LineReset,0,0,100,202109260132500,20,0,33[CR]

受信文字列: OK,C6[CR]

NG,C1[CR] (リセット動作中の場合、NG が返る)

- **締め処理開始コマンド (StartClosing) パラメータ数:1、応答データ数:0**

締め処理を開始します。

パラメータ:

param1: 清算時刻データ(15文字)

送信文字列(例): StartClosing,202109260132500,26[CR]

受信文字列: OK,C6[CR]

NG,C1[CR] (アイドル状態以外の場合、NGが返る)

- **締めデータ読出しコマンド(ReadData)パラメータ数:0、応答データ数:6**

締め処理時、前回の GetStatus 後に読み出した印字データを返します。

送信文字列 : ReadData,22[CR]

受信文字列(例): OK,{16文字},{16文字},{16文字},{16文字},{16文字},{16文字},xx[CR]

BUSY,6F[CR] 印字データ準備中

NG,C1[CR] (NG時にはデータを返さない)

応答データ:

data1: 印字データ1(半角英数カナ16文字)

data2: 印字データ2(半角英数カナ16文字)

data3: 印字データ3(半角英数カナ16文字)

data4: 印字データ4(半角英数カナ16文字)

data5: 印字データ5(半角英数カナ16文字)

data6: 印字データ6(半角英数カナ16文字)

- **結果詳細問い合わせコマンド(GetSettledData)パラメータ数:0、応答データ数:7**

GetStatus で決済結果を取得した後、決済結果の詳細を読み出す。

送信文字列 : GetSettledData,9B[CR]

受信文字列(例): OK,01,01,11,1,3943,3942,*****0512,B4[CR]

NG,C1[CR] (NG時にはデータを返さない)

応答データ:

data1: 決済結果(バイナリー値を ASCII2文字で返す)

data2: 決済コラム(バイナリー値を ASCII2文字で返す)

data3: 決済種別(バイナリー値を ASCII2文字で返す)

data4: モバイル決済額(ASCII 数値 ゼロサプレス)

data5: 決済前モバイル残高(ASCII 数値 ゼロサプレス)

data6: 決済後モバイル残高(ASCII 数値 ゼロサプレス)

data7: モバイルID(ASCII 文字列16文字)

- レシートデータ取得コマンド(*GetReceiptData*)パラメータ数:0、応答データ数:5

レシート印字有効時、GetStatus で決済結果を取得した後、レシート印字データを読み出す。

送信文字列 : GetReceiptData,9B[CR]

受信文字列(例):

```
OK,01,01,07,01,[01][14]*****00000000337[02][10]*****0512[10][0  
E]20221021214002[06][06]920332[07][09]000000285[08][01]1[09][04]3941,A1 [CR]
```

([]で囲われたデータは ASCII 制御コード)

NG,C1[CR] (NG 時にはデータを返さない)

応答データ:

data1: 送信データ種別(BCD2桁を ASCII2文字で返す)

data2: ブランド情報1(BCD2桁を ASCII2文字で返す)

data3: ブランド情報2(ASCII2文字)

data4: レシート種別(BCD2桁を ASCII2文字で返す)

data5: IM10 から送られてくる印字パラメータ(ASCII コード 最大128文字)

- バージョン情報取得コマンド(*GetVersion*)パラメータ数:0、応答データ数:1

BVCM-MIF-02 のファームウェアバージョン文字列を返します。

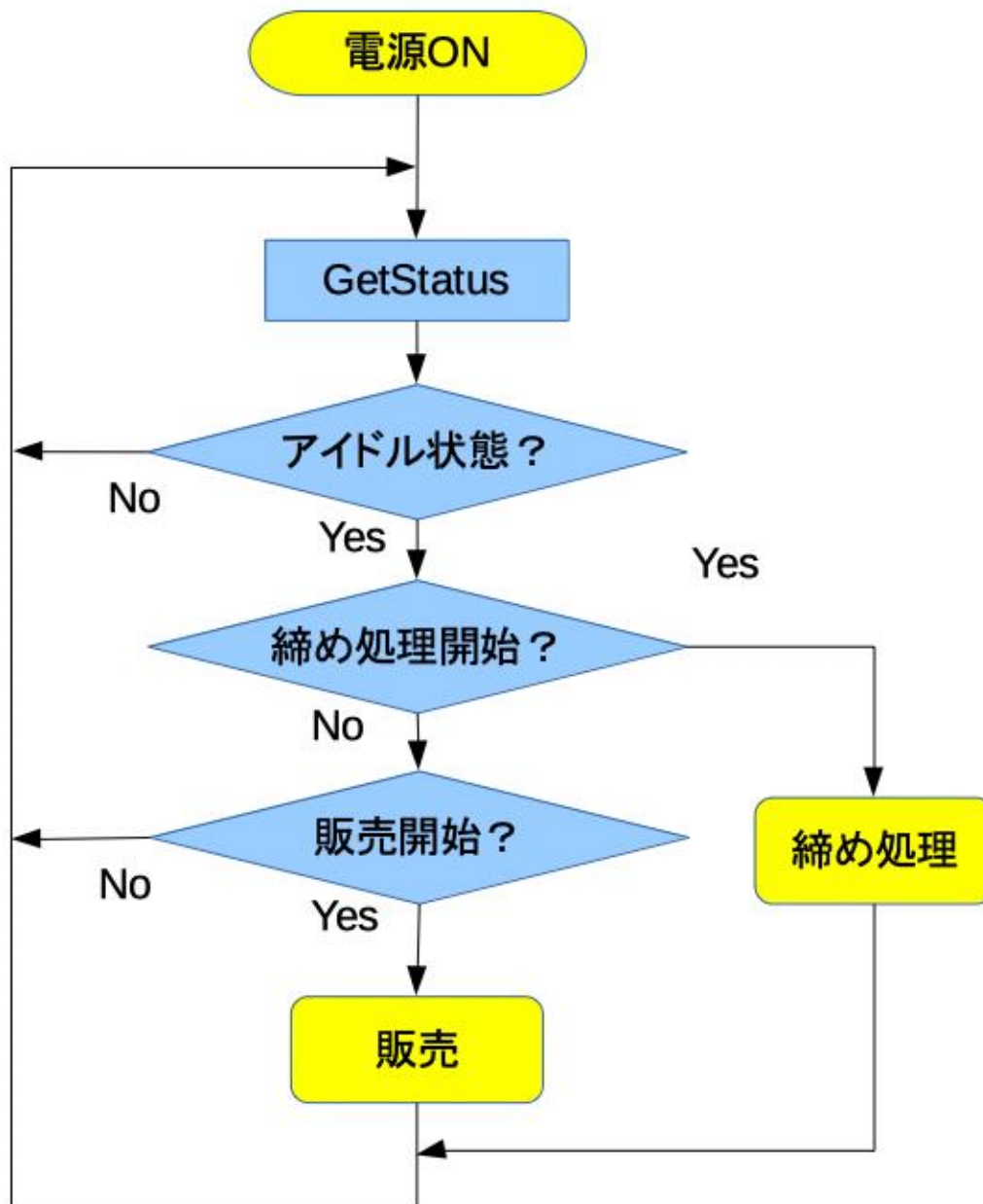
送信文字列: GetVersion,32[CR]

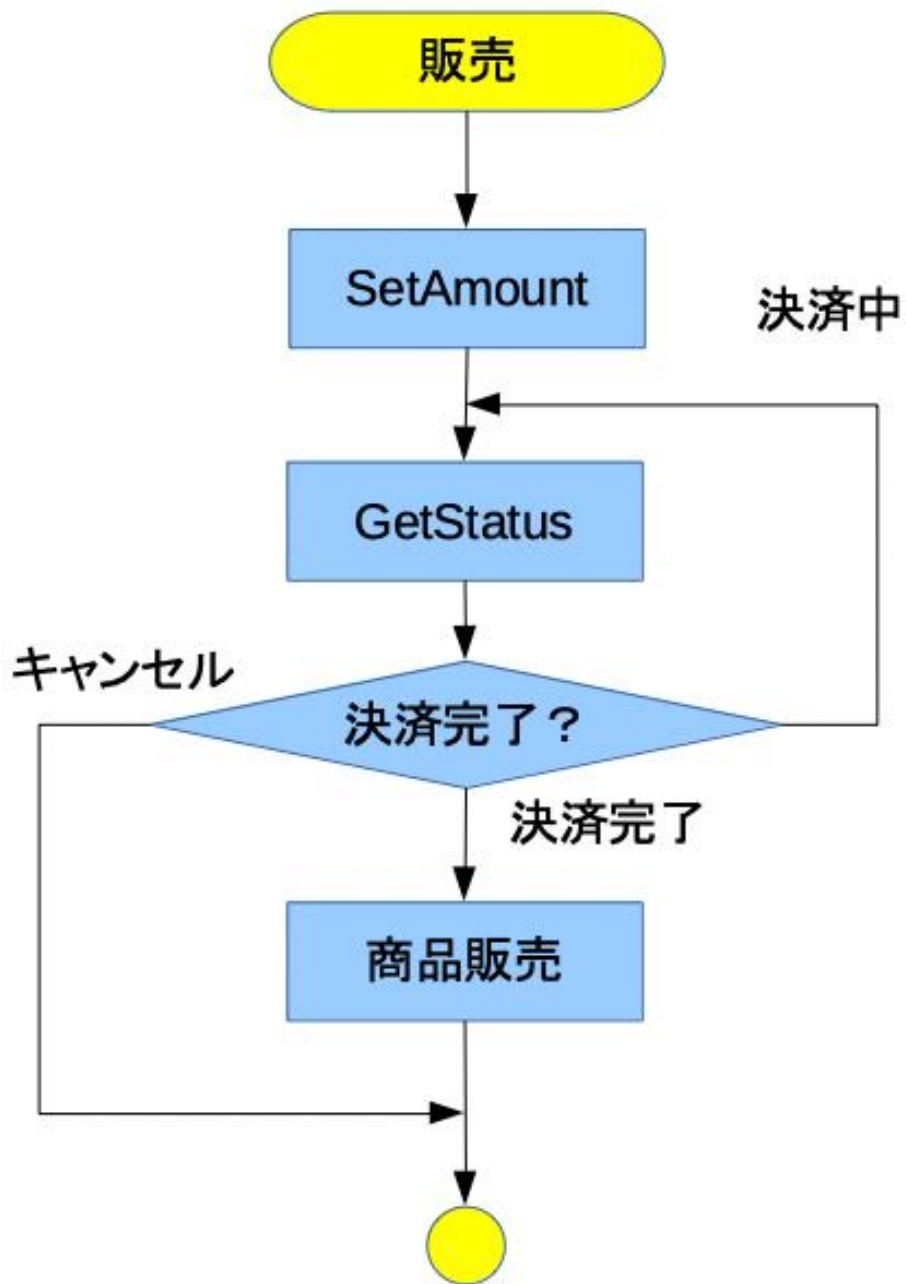
受信文字列: バージョン文字列, チェックサム[CR]

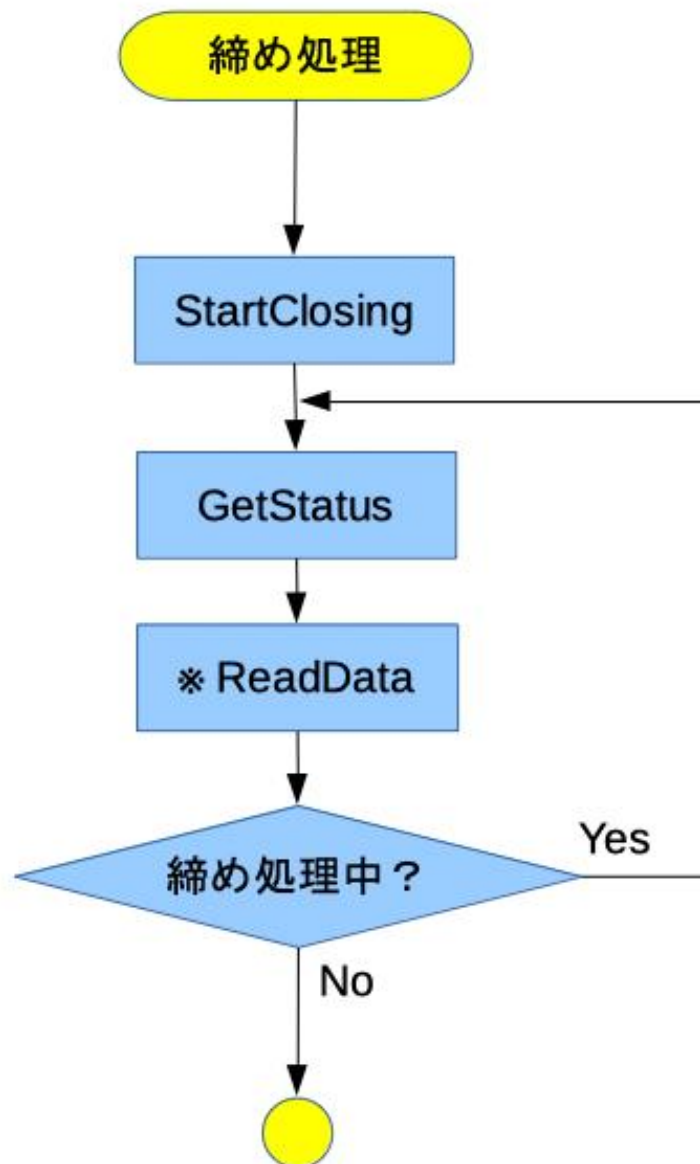
§ 6 動作フロー

- ホストはいつでも GetStatus コマンドを発行してステータスを確認する事が出来ます。BVCM-MIF-02-IM10 は GetStatus コマンドにより駆動され状態遷移しますので、如何なる状態下でも 500msec.以内に1回、GetStatus コマンドを発行し続ける必要があります。
- ステータスがアイドル状態であれば、決済処理あるいは締め処理を行う事が出来ます。
- アイドル状態で SetAmount コマンドを発行し決済金額を設定すると IM10 は決済状態となり、カード受付状態となります。
- 次に IM10 が電子マネーを受け付け、決済完了すると、処理完了状態となります。
- 決済中、決済端末のキャンセルボタンあるいはキャンセルコマンドによりキャンセル状態となった場合、直ちに処理完了状態となります。(IM10 はキャンセル機能はありません。)
- アイドル状態で StartClosing コマンドを発行するあるいは、IM10 の操作により締め処理が開始されると締め処理中状態となります。
- 締め処理は GetStatus コマンドを発行し続ける事により締め処理が進行します。GetStatus を一回発行する毎に締め印字データが IM10 より送られますので、GetStatus コマンドが終了した後、ReadData コマンドを発行して締め印字データを取得する事が出来ます。
- ReadData コマンドはオプションです。データが不要であれば発行せずに次の GetStatus コマンドを送る事が出来ます。
- 締め処理が終了すると、ステータスがアイドルに戻ります。

制御フローチャート







※ ReadDataはオプション。印字データが不用であれば Callする必要はない。