
FOMA UART インタフェースを利用するための技術参考資料

第1.1版

2008/1/25

株式会社エヌ・ティ・ティ・ドコモ

目 次

1	はじめに	4
2	参考資料	4
3	語彙・略号	4
4	物理インタフェース	5
4.1	機器構成	5
4.2	コネクタ形状	5
4.3	携帯電話の状態定義	5
4.4	端子配置	6
4.4.1	外部機器接続時の端子配置.....	6
4.4.2	各端子の役割.....	6
4.5	電气的条件	7
4.5.1	入出力レベル.....	7
4.5.2	入出力等価回路.....	7
4.6	データフォーマット	8
5	UART機能の動作規定	8
5.1	iアプリを利用したUART経由での外部機器とのシリアル通信	8
5.1.1	UARTによるコネクション.....	8
5.1.2	UART によるコネクションのオープン.....	9
5.1.3	UART通信におけるデータ送受信の挙動.....	12
5.1.4	準正常動作.....	14
5.1.5	競合動作.....	14
5.1.6	UART通信中断時の動作.....	14
6	機器識別信号	15
6.1	メッセージ	15
6.1.1	メッセージフォーマット.....	15
6.1.2	メッセージの詳細.....	16
6.2	外部機器側の注意事項	19
7	シーケンス例	20
7.1	iアプリを利用したUART機能	20

7.1.1	外部機器接続後にiアプリ起動	20
7.1.2	機器拒否の送信動作	21
7.1.3	UART接続のクローズ動作	22
7.1.4	iアプリのサスペンド動作	23
7.1.5	iアプリの終了動作	24
7.1.6	バッファオーバーフロー動作	25
7.1.7	機器識別要求後のiアプリ サスペンド動作	26
7.1.8	UART クローズ後のUART再オープン動作	27
8	<i>iアプリから利用する関数</i>	28

1 はじめに

本ドキュメントでは、FOMA端末と外部機器との間をUARTインタフェースで接続する機能について記載する。本機能により、外部機器とシリアル通信が可能となる。

本ドキュメントはiアプリを利用したUART機能を主に規定するが、各FOMA端末に実際に搭載されているサービスの有無により、本ドキュメントに記載されている機能の全部、または一部が使用できない場合がある。また、本ドキュメントに記載された内容は、今後の標準化や機能追加により変更される可能性がある。なお、UART機能はARIB Aコネクタを使用するが、端子条件はUART通信用に規定されるため、USB接続ではないことに十分留意すること。

2 参考資料

本ドキュメントにて参照している仕様書等を以下に示す。

- [1] ARIB TR-T12-27.A01 Ver.3.1.0
- [2] FOMA USB インタフェースを利用するための技術参考資料
- [3] Java 2 Platform, Micro Edition (J2ME) Connected Limited Device Configuration (CLDC) specification (<http://java.sun.com/products/cldc/>)

3 語彙・略号

本ドキュメントで使用する略語・記号を以下に示す。

ARIB : Association of Radio Industries and Businesses

ME : Mobile Equipment

UART : Universal Asynchronous Receiver Transmitter

外部機器 : UART-IF 経由で FOMA 端末と通信する機器

受信バッファ : 外部機器から受信したデータを i アプリが InputStream から read()メソッドを呼び出すまでの間、(k)VM が一時的に格納するバッファ

送信バッファ : i アプリが OutputStream から write()メソッドにより UART へ入力したデータを外部機器に送信するまでの間、(k)VM が一時的に格納するバッファ

アプリ UART : i アプリを利用して行う UART 通信

4 物理インタフェース

4.1 機器構成

本仕様における基本的な機器構成を図 4-1に示す。

なお、外部機器との接続手段によっては、外部機器付属ケーブルが無い場合もある。

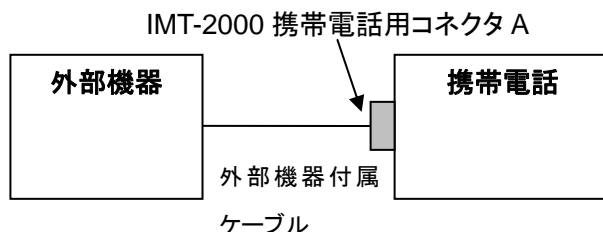


図 4-1 機器構成

4.2 コネクタ形状

携帯電話側で使用するARIB Aコネクタ(IMT-2000 携帯電話用コネクタA)のコネクタ形状略図を図 4-2に示す。詳細は、[1]を参照。

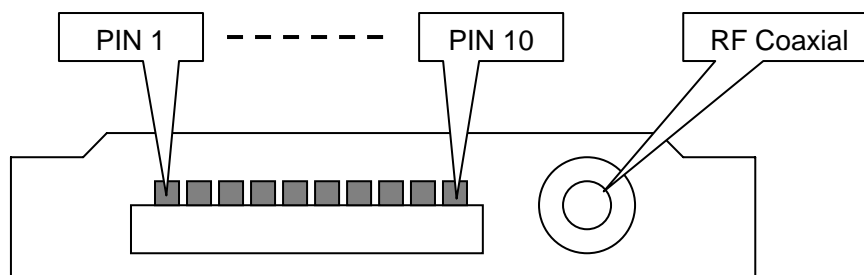


図 4-2 コネクタ形状略図

4.3 携帯電話の状態定義

Manufacturer Specific、VBusの両端子の端子条件に応じて、ARIB Aコネクタで提供する機能を表 4-1に示す。なお、携帯電話が電源ON状態の場合に、Manufacturer Specific端子を検出することとし、携帯電話が電源OFF状態の場合には検出する必要はない。

また、本ドキュメントで規定する動作を行う場合、USB接続で用いるコネクタを共用するが、表 4-2に示すUART用の端子配置になるため、USB接続とはならない。

表 4-1 端子配置による携帯電話の状態定義

		Manufacturer Specific(8)	
		Hi	Low
USB VBus (4)	5Vin	[2]に従うこと。	[2]に従うこと。
	Hi(*)	(未接続)	本ドキュメントで規定する動作を行うこと。

(*)開放相当の high インピーダンス

4.4 端子配置

4.4.1 外部機器接続時の端子配置

外部機器接続時のコネクタの端子配置を表 4-2に示す。

表 4-2 外部機器接続時の端子配置

Pin No	NAME	Direction		NOTE
		外部機器	携帯電話	
1	GND (UART GND)	---		
2	RXD	>>>		
3	TXD	<<<		
4	---	Not Connected		
5	Power Supply	>>>		
6	Reserved	Not Defined		
7	RTS	<<<		フロー制御時に使用
8	Manufacturer Specific	>>>		
9	CTS	>>>		フロー制御時に使用
10	GND	---		
RF	RF TRX	<<<>>>		Coaxial

4.4.2 各端子の役割

表 4-2にある端子の役割を述べる。

(1) TXD

携帯電話から外部機器へデータを送信するための端子。

(2) RXD

外部機器から送られてくるデータを受信するための端子。

(3) TXD/RXD GND

データ送受信の接地端子。

(4) RTS

RXD のフロー制御用の端子。フロー制御を行う場合、データを受信できない時に Hi 状態とすること。

(5) CTS

TXD のフロー制御用の端子。フロー制御を行う場合、Low 状態の時に通信を行うこと。

(6) Manufacturer Specific

携帯電話が外部機器の接続を検出するための端子。

4.5 電氣的条件

4.5.1 入出力レベル

信号の入出力レベルを表 4-3に示す。

表 4-3 デジタル信号の電氣的条件

Pin No	端子名	Hi レベル	Low レベル
2	RXD	開放相当の high インピーダンス	0~0.6V
3	TXD	開放相当の high インピーダンス	0~0.6V (シンク電流 0.5mA 以下、 電源電圧 2V 時) (シンク電流 1.0mA 以下、 電源電圧 3.3V 時)
7	RTS	開放相当の high インピーダンス	0~0.6V (シンク電流 0.5mA 以下、 電源電圧 2V 時) (シンク電流 1.0mA 以下、 電源電圧 3.3V 時)
8	Manufacturer Specific	開放相当の high インピーダンス	0~0.6V
9	CTS	開放相当の high インピーダンス	0~0.6V

4.5.2 入出力等価回路

表 4-3の各端子の入出力等価回路を図 4-3に示す。

出力はオープンコレクタ又はオープンドレインとする。

入力はプルアップとし、プルアップ電源は 1.8V 以上 3.3V 以下とする。

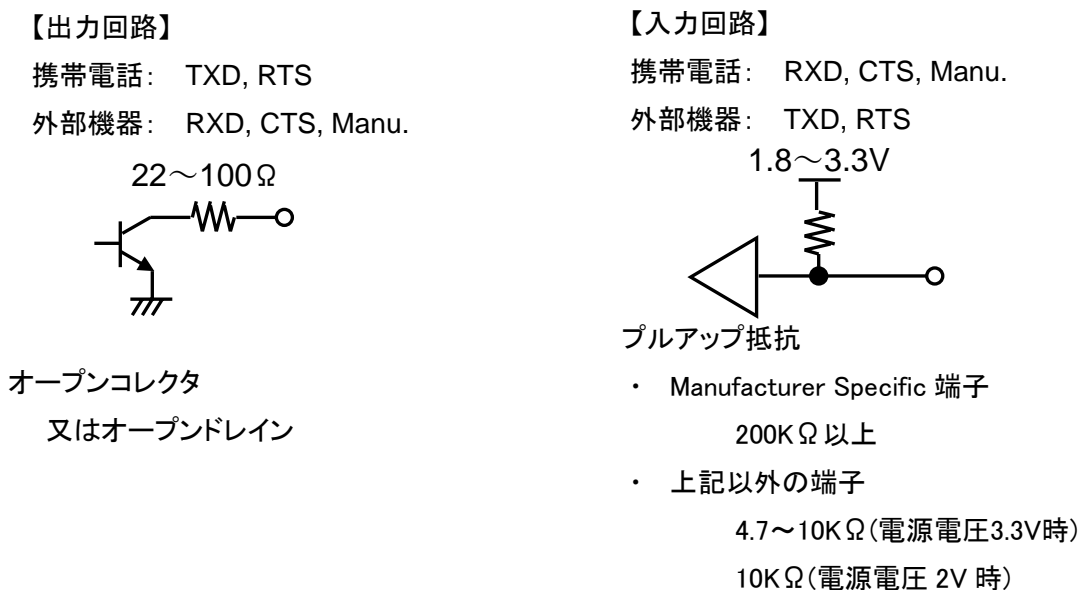


図 4-3 各端子の入出力等価回路

4.6 データフォーマット

本仕様で使用するデータフォーマットは、以下の要件に従う。

- ・ データは調歩同期とし、LSB ファーストで送受信する。
- ・ データフォーマットはデータビット長 8 ビット、パリティなし、ストップビット長 1 ビットに対応する。詳細は 5.1.2.1 章参照。

図 4-4 の例は、スタートビット 1 ビット、データビット長 8 ビット、パリティなし、ストップビット長 1 ビットの合計 10 ビットで構成されるデータフォーマット例である。

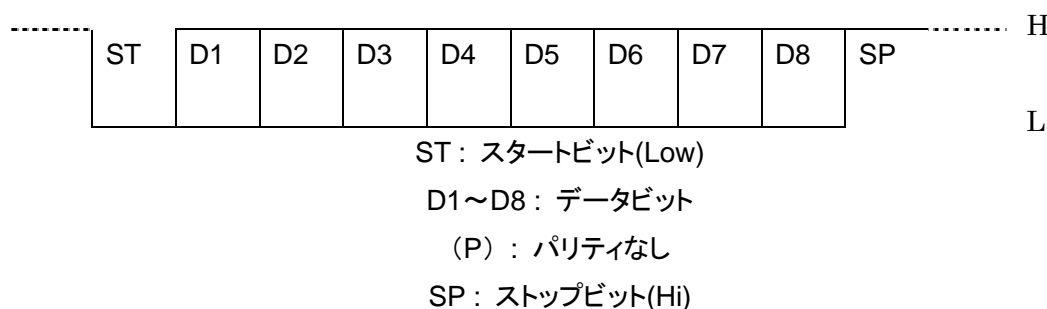


図 4-4 データフォーマット例

5 UART 機能の動作規定

5.1 i アプリを利用した UART 経由での外部機器とのシリアル通信

i アプリを使用し、UART 経由で外部機器とシリアル通信を行う機能は、以下の規定に従う。

5.1.1 UART によるコネクション

UART によるコネクションは、以下の規定に従う。

- ・ URLとパラメータを指定し、Connector.open()メソッドにて、UART によるコネクションをオープンする。詳細は5.1.2章参照。
- ・ UART によるコネクションオープン時にREADモード、WRITEモード、READ_WRITEモードを指定できる。詳細は5.1.2.2章参照。
- ・ FeliCa 機能の利用とは排他扱いとし、SDF の UseFeliCaOnline キー、SDF の UseFeliCaOffline キー、ADF の FeliCaCPID キー、ADF の FeliCaPIN キーの、いずれかが指定されているときにUARTによるコネクションをオープンした場合は、SecurityExceptionが発生する。
- ・ 1つのスレッド、または、複数のスレッドから UART によるコネクションの多重オープンしようとした場合、ConnectionException(BUSY_RESOURCE)が発生する。

5.1.2 UART による接続のオープン

UART による接続をオープンする場合に指定する URL は次の様に指定し、n は通信ポートの番号とする。

```
open (comm:/n )
```

ただし、UART による接続では通信ポートは一つしか利用できないため、n は 0 で固定となる。n に 0 以外が指定された場合、IllegalArgumentException が発生する。

また、オプションのパラメータとして、UART による接続 オープン時のボーレート(baudrate)・データビット長(databitlen)・パリティ(parity)・ストップビット長(stopbitlen)・フロー制御(flowctrl)を設定でき、次のように指定する。

```
open(comm:/0;baudrate=<整数>,databitlen=<整数>,parity=<文字列>,
stopbitlen=<文字列>,flowctrl=<文字列>)
```

設定したパラメータの有効値は携帯電話の実装に依存する。詳細は5.1.2.1章参照。

5.1.2.1 オプションのパラメータに指定可能な文字

UART による接続をオープンする際は、以下の規定に従う。

- ・ ボーレート、データビット長、パリティ、ストップビット長、およびハードウェアフロー制御を指定する際、その表記 (baudrate=<整数>,databitlen=<整数>,parity=<文字列>,stopbitlen=<文字列>,flowctrl=<文字列>) は大文字/小文字のどちらでもよい。
- ・ 各パラメータの指定順序は問わない。
- ・ name に baudrate, databitlen, parity, stopbitlen, flowctrl のいずれかが指定され、value が有効な指定値でない場合(詳細は次章参照)、IllegalArgumentException が発生する。
- ・ 同一パラメータが複数回指定された場合、指定された値が有効な値か無効な値かに関わらず、後から指定された値が設定値として扱われる。
- ・ 各パラメータは全てオプションとし、一部のパラメータを指定するだけでもよい。指定のないパラメータは次が指定されたものとして扱う。

ボーレートは “9600”、データビット長は “8”、パリティは “NO”、ストップビット長は “1”、ハードウェアフロー制御は “OFF” 。

5.1.2.1.1 ボーレート

ボーレートは baudrate=<整数>で指定する。ボーレートを指定する際、“9600”、“38400”、“115200”を有効な指定値とする。左記以外を指定した場合、有効な指定値として扱われるかは携帯電話の実装に依存する。

指定値の意味を表 5-1に示す。

表 5-1 ボーレートの指定値

値	意味
9600	9600 bps
38400	38400 bps
115200	115200 bps

5.1.2.1.2 データビット長

データビット長はdatabitlen=<整数>で指定する。データビット長を指定する際、“8”を有効な指定値とする。左記以外を指定した場合、有効な指定値として扱われるかは携帯電話の実装に依存する。指定値の意味を表 5-2に示す。

表 5-2 データビット長の指定値

値	意味
8	8 ビット

5.1.2.1.3 パリティ

パリティはparity=<文字列>で指定する。パリティを指定する際、“NO” を有効な指定値とする。“ODD”, “EVEN”, “MARK”, “SPACE” が有効な指定値として扱われるかは携帯電話の実装に依存する。指定値の意味は表 5-3に示す。

表 5-3 パリティの指定値

値	意味
NO	パリティなし
ODD	奇数パリティ
EVEN	偶数パリティ
MARK	常にパリティビットとして'1'を付加
SPACE	常にパリティビットとして'0'を付加

5.1.2.1.4 ストップビット長

ストップビット長はstopbitlen=<文字列>で指定する。ストップビット長を指定する際、“1” を有効な指定値とする。左記以外(“1.5”, “2”など)を指定した場合、有効な指定値として扱われるかは携帯電話の実装に依存する。指定値の意味は表 5-4に示す。

表 5-4 ストップビット長の指定値

値	意味
1	1 ビット
1.5	1.5 ビット
2	2 ビット

5.1.2.1.5 フロー制御

本ドキュメントにおけるフロー制御はすべて、ハードウェアフロー制御とする。

フロー制御はflowctrl=<文字列>で指定する。ハードウェアフロー制御を指定する際、“OFF” を有効な指定値とする。“ON” が有効な指定値として扱われるかは携帯電話の実装に依存する。指定値の意味は表 5-5に示す。

表 5-5 フロー制御の指定値

値	意味
OFF	ハードウェアフロー制御を行わない
ON	ハードウェアフロー制御を行う

5.1.2.2 UART のオープンモード

UART により双方向のシリアル通信を行う場合は READ_WRITE モードを指定して UART のコネクションをオープンする。

UART を READ_WRITE モードでオープンした場合は InputConnection インタフェースと OutputConnection インタフェースの両方に定義されているメソッドを呼び出すことができる。

また、READ_WRITE モードでオープンしたコネクションから OutputStream と InputStream を同時に取得できる。

5.1.3 UART 通信におけるデータ送受信の挙動

5.1.3.1 UART がオープンされている場合

i アプリの open() メソッドにより UART がオープンされている場合の挙動を記載する。

- ・ 外部機器から受信したデータは受信バッファに格納する。(*1)
- ・ i アプリが read() メソッドを呼び出し、受信バッファからデータを取り出す。
- ・ i アプリが write() メソッドを呼び出し、i アプリからのデータを送信バッファに格納する。(*1)
- ・ 送信バッファに格納されたデータは、格納され次第速やかに外部機器へ送信される。
- ・ i アプリが flush() メソッドを呼び出すと、送信バッファに格納されたデータがすべて外部機器へ送信される。
- ・ i アプリが available() メソッドを呼び出すと、受信バッファに格納されている容量(Byte)が返る。
- ・ InputStream クラスの mark() メソッド、reset() メソッドへの対応は携帯電話の実装に依存する。
- ・ i アプリ上で書き込み・読み込み可能なデータサイズは、携帯電話の実装に依存する。
- ・ InputStream または OutputStream の多重オープンをしようとした場合、IOException が発生する。ただし、READ_WRITE モードの場合、InputStream と OutputStream は同時にオープンできる。
- ・ 携帯電話と外部機器のボーレートが一致せずデータを正しく読み取れない場合は、そのデータは破棄される。
- ・ パリティが存在する場合において、外部機器から受信したデータのパリティチェックが NG の場合、そのデータは破棄される。
- ・ read したデータのストップビットとして Low を検出したときには、不正なデータと判断し UART が close される。

*1: 送受信バッファの最大容量は実装依存とする。

5.1.3.2 i アプリが UART によるコネクションをクローズした場合

i アプリから close() メソッドの呼び出しにより UART によるコネクションがクローズされた場合は、以下の動作を行う。

- ・ 受信バッファ内にあるデータはすべて破棄される。
- ・ 送信バッファ内にあるデータは破棄されず、外部機器に送信される。
- ・ コネクションを正常に例外発生させずにクローズできる場合、別スレッドから close() メソッドを呼び出しても例外発生せずにクローズされる。

i アプリは close() メソッドを呼び出す前に flush() メソッドを呼び出す作りにすること。

5.1.3.3 i アプリがサスペンドした場合

i アプリがユーザ操作や音声着信等によりサスペンドした場合、以下の動作に従う。

- ・ 送受信バッファ内にあるデータはすべて破棄される。
- ・ UART がクローズする。
- ・ サスペンド以降に外部機器から受信したデータは破棄される。
- ・ UART 通信中にサスペンドが起こった場合も、上記の動作に従う。(サスペンド時に UART 通信が切断され InterruptedIOException が発生する。)

5.1.3.4 i アプリが終了した場合

i アプリが終了した場合、以下の動作に従う。

- ・ 送受信バッファ内にあるデータはすべて破棄される。
- ・ i アプリ終了以降に外部機器から受信したデータは破棄される。

5.1.3.5 待受 i アプリが状態遷移した場合

5.1.3.5.1 活性化状態から非活性化状態に遷移した場合

待受 i アプリが活性化状態から非活性化状態に遷移した場合、以下の動作に従う。

- ・ 受信バッファ内にあるデータは破棄されず、i アプリに送信される。
- ・ 送信バッファ内にあるデータは破棄されず、外部機器に送信される。
- ・ 非活性化状態に遷移した後、外部機器とのデータ送受信を行う際に再オープンする必要はない。

5.1.3.5.2 非活性化状態から活性化状態に遷移した場合

待受 i アプリが非活性化状態から活性化状態に遷移した場合、以下の動作に従う。

- ・ 受信バッファ内にあるデータは破棄されず、i アプリに送信される。
- ・ 送信バッファ内にあるデータは破棄されず、外部機器に送信される。
- ・ 活性化状態に遷移した後、外部機器とのデータ送受信を行う際に再オープンする必要はない。

5.1.3.5.3 非活性化状態から休眠状態に遷移した場合

待受 i アプリが非活性化状態から休眠状態に遷移した場合、以下の動作に従う。

- ・ 送受信バッファ内にあるデータはすべて破棄される。
- ・ UART がクローズする。
- ・ 休眠状態に遷移した後に外部機器から受信したデータは破棄される。
- ・ UART 通信中に非活性化状態から休眠状態に遷移した場合も、上記の動作に従う。(休眠状態遷移時に UART 通信が切断され InterruptedIOException が発生する。)

5.1.3.6 UART がオープンされていない場合

- ・ UART がオープンされていない場合は、外部機器から受信したデータは破棄される。
- ・ UART がオープンされてから 100ms 以内に、外部機器からのデータを受信できる状態になる。
- ・ UART がオープンされていない場合に InputStream/OutputStream クラスのメソッドを実行した場合、IOException が発生する。

5.1.3.7 i アプリが起動していない場合

i アプリが起動していない場合は、外部機器から受信したデータは破棄される。

5.1.4 準正常動作

5.1.4.1 ケーブル抜け

外部機器と携帯電話を繋いでいる UART ケーブルを抜去した場合、以下の動作に従う。

また、携帯電話が検出している端子条件が表 4-2 で定義される条件から変化した場合も同様とする。

- ・ UART がクローズする。
- ・ 送受信バッファ内にあるデータはすべて破棄される。
- ・ UART 通信中であった場合、Connection Exception(UART_DISCONNECTED)が発生する。

5.1.4.2 バッファオーバーフロー

送受信バッファがいっぱいである場合は、新たに受信したデータは破棄される。

5.1.5 競合動作

5.1.5.1 携帯電話本体動作

外部機器と携帯電話が接続している場合においても、携帯電話本体の操作には影響を与えない。

5.1.6 UART 通信中断時の動作

UART 通信が中断した場合は以下の動作に従う。

- ・ 通信の中断による例外が発生した場合、通信に関わる全てのネイティブリソースが解放された状態から処理が開始される。
- ・ 下位レイヤで通信を行う通信メソッドが正常に完了した後に、通信が中断した場合、通信の中断による例外が発生するまでは、通信の中断前に取得したデータを利用できる。

6 機器識別信号

UART 通信を行う i アプリと外部機器の組み合わせを制限することが望ましい場合がある。その制限手段として機器識別信号を利用する際の、メッセージのフォーマットを例示する。

6.1 メッセージ

6.1.1 メッセージフォーマット

メッセージのフォーマットを表 6-1 に示す。メッセージは「ヘッダ」「種別」「継続 bit」「データ長」「データ」から構成される。

表 6-1 メッセージフォーマット

D8	D7	D6	D5	D4	D3	D2	D1
ヘッダ							
種別							
継続 bit	データ長						
データ							

6.1.1.1 ヘッダ(1byte)

- (1) 本仕様における論理インタフェースに従うメッセージは、必ず当該「ヘッダ」を先頭にする。
- (2) 「ヘッダ」はヘキサで「8C」とする。

6.1.1.2 種別

「種別」はメッセージの役割を表す。種別一覧を表 6-2 および表 6-3 に示す。

表 6-2 オペレータオプションメッセージ

種別	D8	D7	D6	D5	D4	D3	D2	D1	参照
オペレーションオプションメッセージ	1	0	1	1	X	x	x	x	6.1.2.1
機器識別応答(アプリ UART)	1	0	1	1	0	0	0	1	
機器識別要求(アプリ UART)	1	0	1	1	0	0	1	0	

表 6-3 機器識別メッセージ

種別	D8	D7	D6	D5	D4	D3	D2	D1	参照
機器識別メッセージ	1	1	0	0	X	x	x	x	6.1.2.2
機器受付	1	1	0	0	0	0	1	0	
機器拒否	1	1	0	0	0	0	1	1	

6.1.1.3 データ長、継続 bit(1byte)

- (1) メッセージ中の「データ長」は、当該メッセージの「データ」のみの長さを示す。(すなわち「ヘッダ」「種別」「データ長」の長さは含まない)
- (2) 一つの識別メッセージの「データ」が 128oct.以上のメッセージは、以下のように送出する。
1. 「継続 bit」=「継続有り」、「データ長」=127 とし、データの先頭から 127oct.までを「データ」として送信する。
 2. 1の処理後、送信せずに残ったデータの長さが再度 128oct.以上ならば、1.に従って送信する。その際、同じ「種別」を用いて送信する。
 3. 1および2の処理後、残ったデータ長が 127oct.以下ならば、継続bitを「継続無し」、「データ長」=残りのデータの長さとして、残りを送信する。
- (3) 「データ」が 127oct.以下のメッセージは分割しない。
- (4) 「継続bit」のビットフォーマットを表 6-4に示す。

表 6-4 継続 bit

内容	D8
継続無し	0
継続有り	1

6.1.2 メッセージの詳細

6.1.2.1 オペレータオプションメッセージ

オペレータオプションメッセージの機器識別応答(アプリ UART)および機器識別要求(アプリ UART)のフォーマットを以下に示す。

表 6-5 オペレータオプションメッセージのフォーマット

種別	D8 D7 D6 D5 D4 D3 D2 D1	備考
オペレータ オプション メッセージ (機器識別 応答(アプリ UART))	1 0 0 0 1 1 0 0	ヘッダ(8C)
	1 1 0 0 0 1 0 1	種別(オペレータオプションメッセージ(機器識別応答(アプリ UART)))
	0 0 0 1 1 0 0 0	データ長 24
	X8 X7 X6 X5 X4 X3 X2 X1	(X8 X7 X6):速度値(表 6-6) (X5 X4 X3):パリティ値(表 6-7) (X2 X1):ストップビット長(表 6-8)
	Y8 Y7 0 0 0 0 0 0	Y8=0/1 :フロー制御無し/あり Y7=0/1 :オペレータ :その他/NTT DoCoMo
	E8 E7 E6 E5 E4 E3 E2 E1	En : 外部機器の製造元 10文字固定(*1)

	F8 F7 F6 F5 F4 F3 F2 F1	Fn : 外部機器の製品名 10文字固定(*1)

	G8 G7 G6 G5 G4 G3 G2 G1	Gn : 外部機器のバージョン 3文字固定(*2)
...	...	
オペレータオ プション メッセージ (機器識別要 求(アプリ UART)) (*3)	1 0 0 0 1 1 0 0	ヘッダ
	1 1 0 0 0 1 0 1	種別(オペレータオプションメッセージ(機器識別要求(アプリ UART)))
	0 0 1 0 0 1 0 0	データ長 36
	E8 E7 E6 E5 E4 E3 E2 E1	En : 携帯電話が対応している本ドキュメントのバージョン 3文字固定(*2)

	F8 F7 F6 F5 F4 F3 F2 F1	Fn : 機種名 10文字固定(*1)

	G8 G7 G6 G5 G4 G3 G2 G1	Gn : iアプリの製造元 10文字固定(*1)

	H8 H7 H6 H5 H4 H3 H2 H1	Hn : iアプリの名称 10文字固定(*1)
...	...	
I8 I7 I6 I5 I4 I3 I2 I1	In : iアプリのバージョン 3文字固定(*2)	
...	...	

*1:英数字で表示させること。また、10 文字以上になる場合は、切り捨て、省略等の処置を行い、10 文字にする。また、10 文字未満になるときは、スペースを用いて 10 文字にする。

*2:バージョンは 3 文字で表示すること。(例 :バージョン 100)

*3:UART がオープンしたタイミングで外部機器に送信する。

表 6-6 速度値

通信速度	X8	X7	X6
600bps	0	0	0
2,400bps	0	0	1
9,600bps	0	1	0
38,400bps	0	1	1
115,200bps	1	0	1
230,400bps	1	1	0
460,800bps	1	1	1

表 6-7 パリティ値

パリティ	X5	X4	X3
なし	0	0	1
偶数パリティ	0	1	0
奇数パリティ	0	1	1
マークパリティ	1	0	0
スペースパリティ	1	0	1

表 6-8 ストップビット長

ストップビット長	X2	X1
1	0	1
1.5	1	0
2	1	1

6.1.2.2 機器識別メッセージ

機器識別メッセージの機器受付、および機器拒否のフォーマットを以下に示す。

表 6-9 機器識別メッセージ(下り)

種別	D8 D7 D6 D5 D4 D3 D2 D1	備考
機器受付	1 0 0 0 1 1 0 0	ヘッダ
	1 1 0 0 0 0 1 0	種別(機器受付)
	0 0 0 0 0 0 0 0	データ長 0
機器拒否	1 0 0 0 1 1 0 0	ヘッダ
	1 1 0 0 0 0 1 1	種別(機器拒否)
	0 0 0 0 0 0 0 1	データ長 1
	R8 R7 R6 R5 R4 R3 R2 R1	Rn:理由値(表 6-10参照)

表 6-10 理由値 (機器拒否)

理由	D8 D7 D6 D5 D4 D3 D2 D1	説明
携帯電話非対応	0 0 0 1 0 0 0 0	携帯電話がこのインタフェース状態に対応していない
速度非対応	0 0 0 1 0 0 1 1	携帯電話がこの速度に対応していない
データ部規定外	0 0 1 0 0 0 0 1	データ部が規定していないパラメータなので処理を実行できない
外部機器想定外	0 0 1 0 0 0 1 0	外部機器が対応していない
その他	1 1 1 1 1 1 1 1	上記以外の理由

6.2 外部機器側の注意事項

外部機器は、以下のことを注意する必要がある。

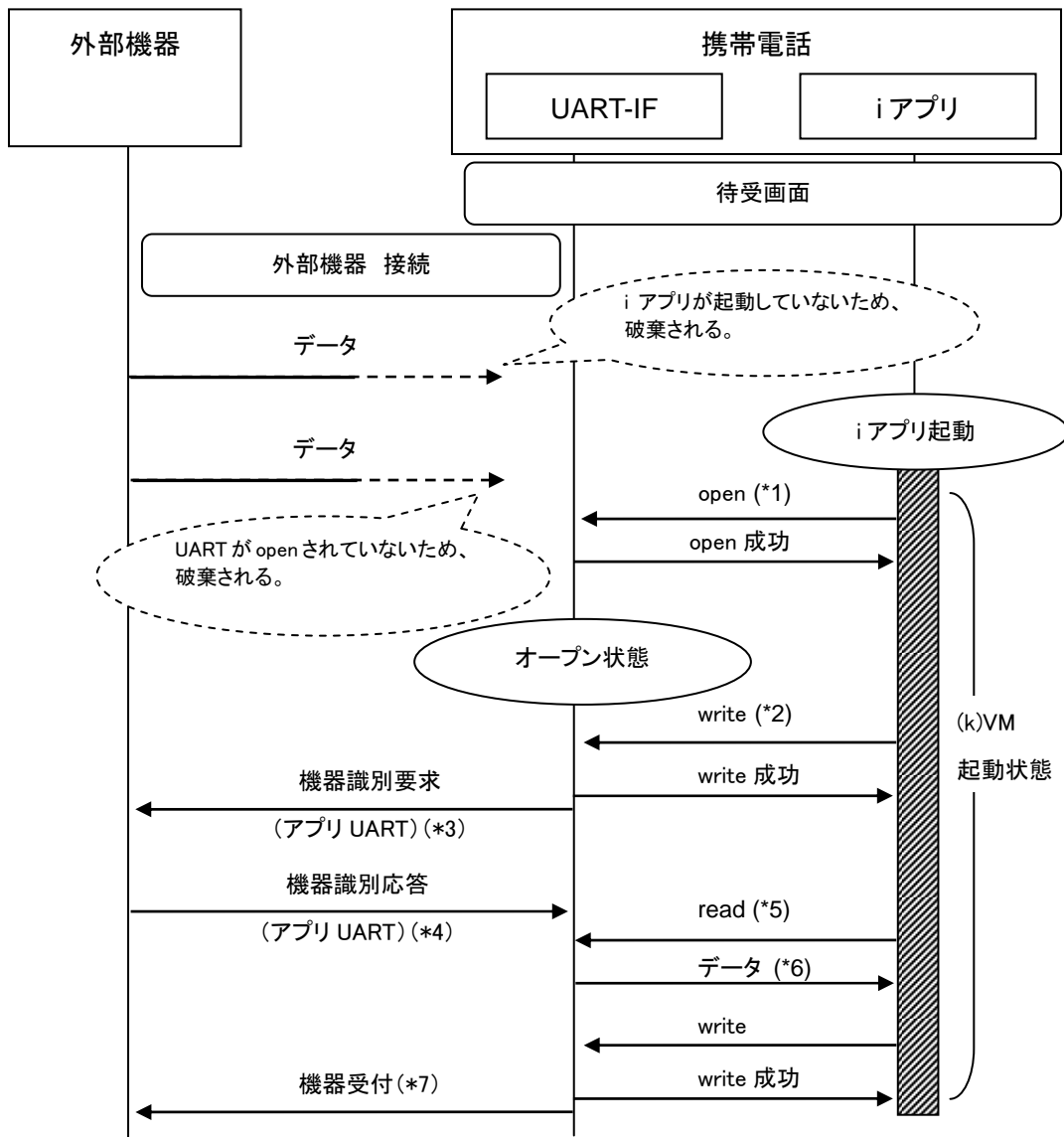
- ・ 機器拒否を送信した場合、UART をクローズすること。
- ・ UARTをクローズし、再オープンした場合に受信する機器識別要求(アプリUART)に対しても、同様に携帯電話に機器識別応答(アプリUART)を送信すること。(7.1.8章のシーケンス参照)

7 シーケンス例

7.1 i アプリを利用した UART 機能

7.1.1 外部機器接続後にi アプリ起動

外部機器接続後にi アプリ起動した場合において、UART 通信を開始するまでのシーケンスを以下に示す。



*1: UART をオープンさせるメソッド。URL のパラメータによりボーレート等を設定する。

*2: 外部機器にデータを送信するメソッド。データを送信バッファに格納する。

*3: 送信バッファに格納されたデータを格納され次第、速やかに外部機器へ送信する。

*4: 外部機器からのデータを受信バッファに格納する。

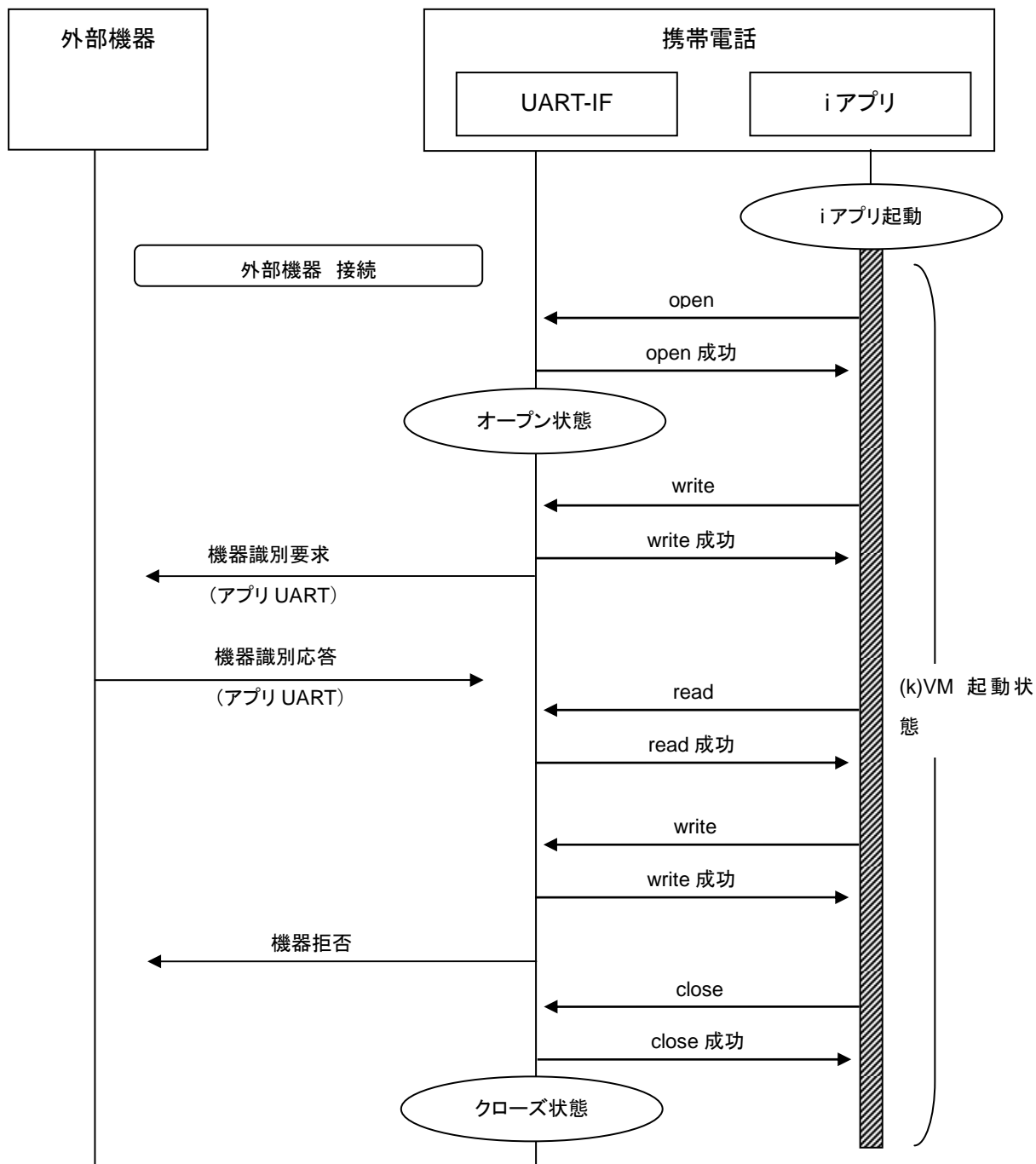
*5: 受信バッファにあるデータを取得するメソッド。

*6: 受信バッファにあるデータを取り出し、i アプリに送信する。

*7: 外部機器が機器受付を受信したら、UART 通信が開始される。

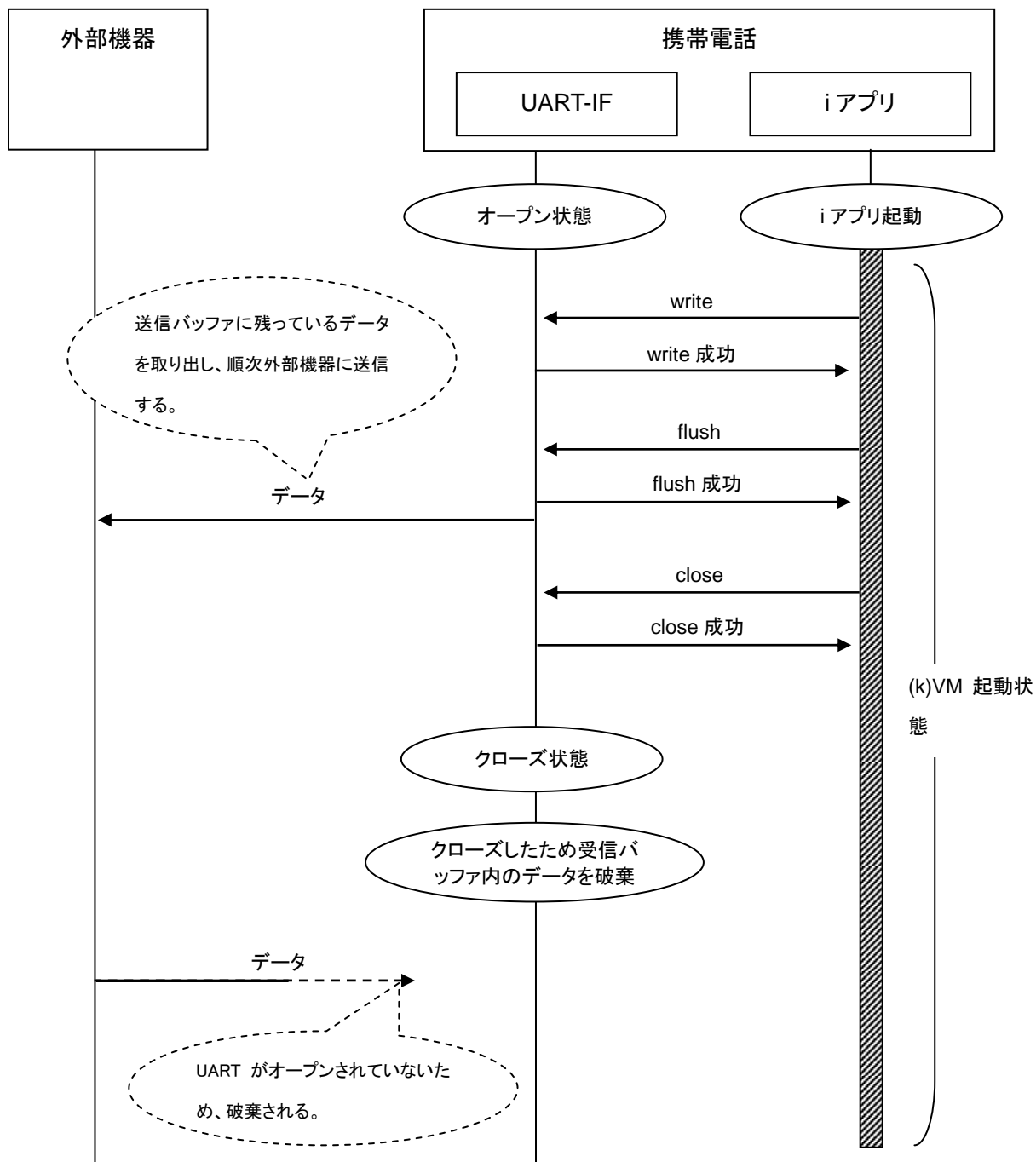
7.1.2 機器拒否の送信動作

外部機器からの機器識別応答に不具合があり、機器拒否が送信される場合のシーケンスを以下に示す。



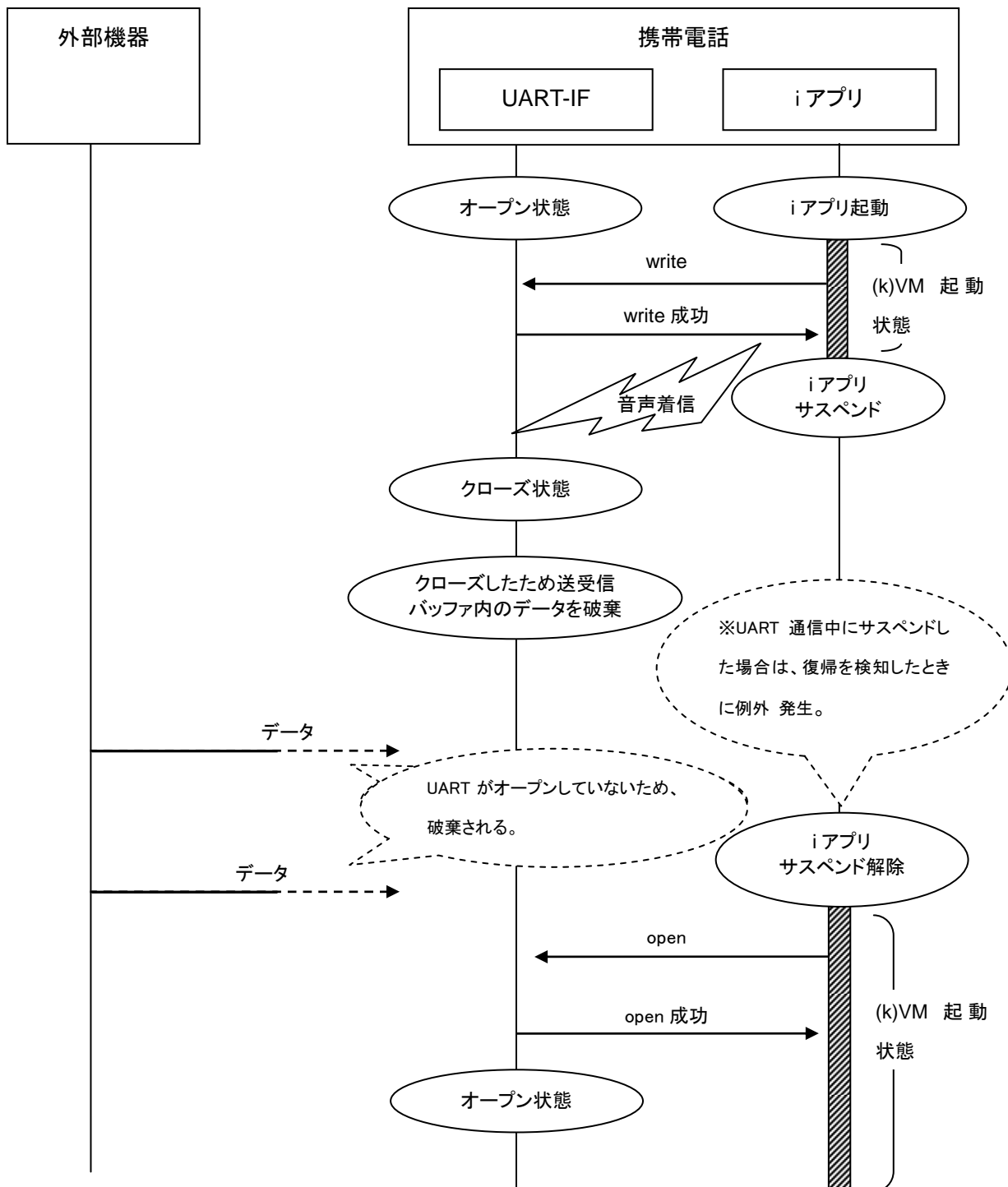
7.1.3 UART コネクションのクローズ動作

UART オープン状態で close()メソッドを呼び出した場合の動作シーケンスを以下に示す。



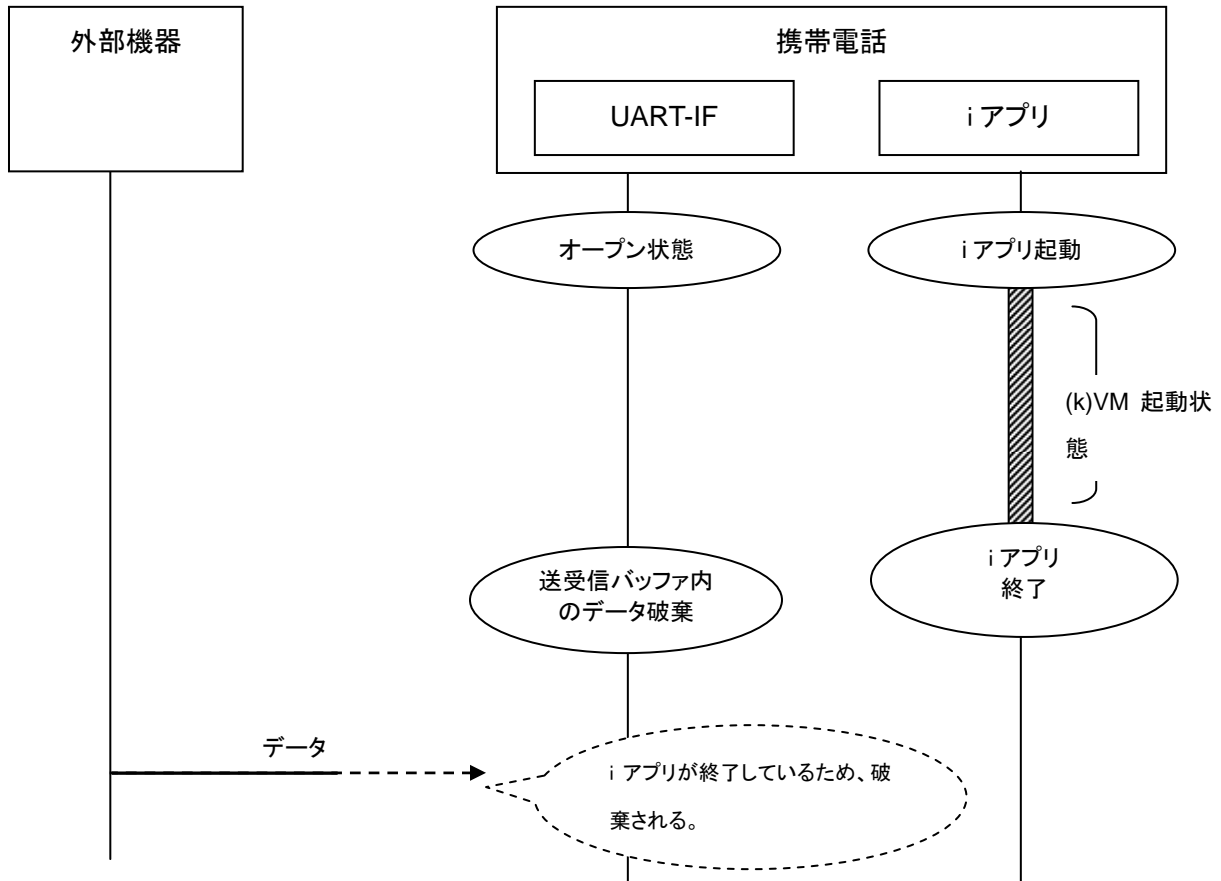
7.1.4 i アプリのサスペンド動作

UART オープン状態で音声着信による i アプリがサスペンドした場合のシーケンスを以下に示す。



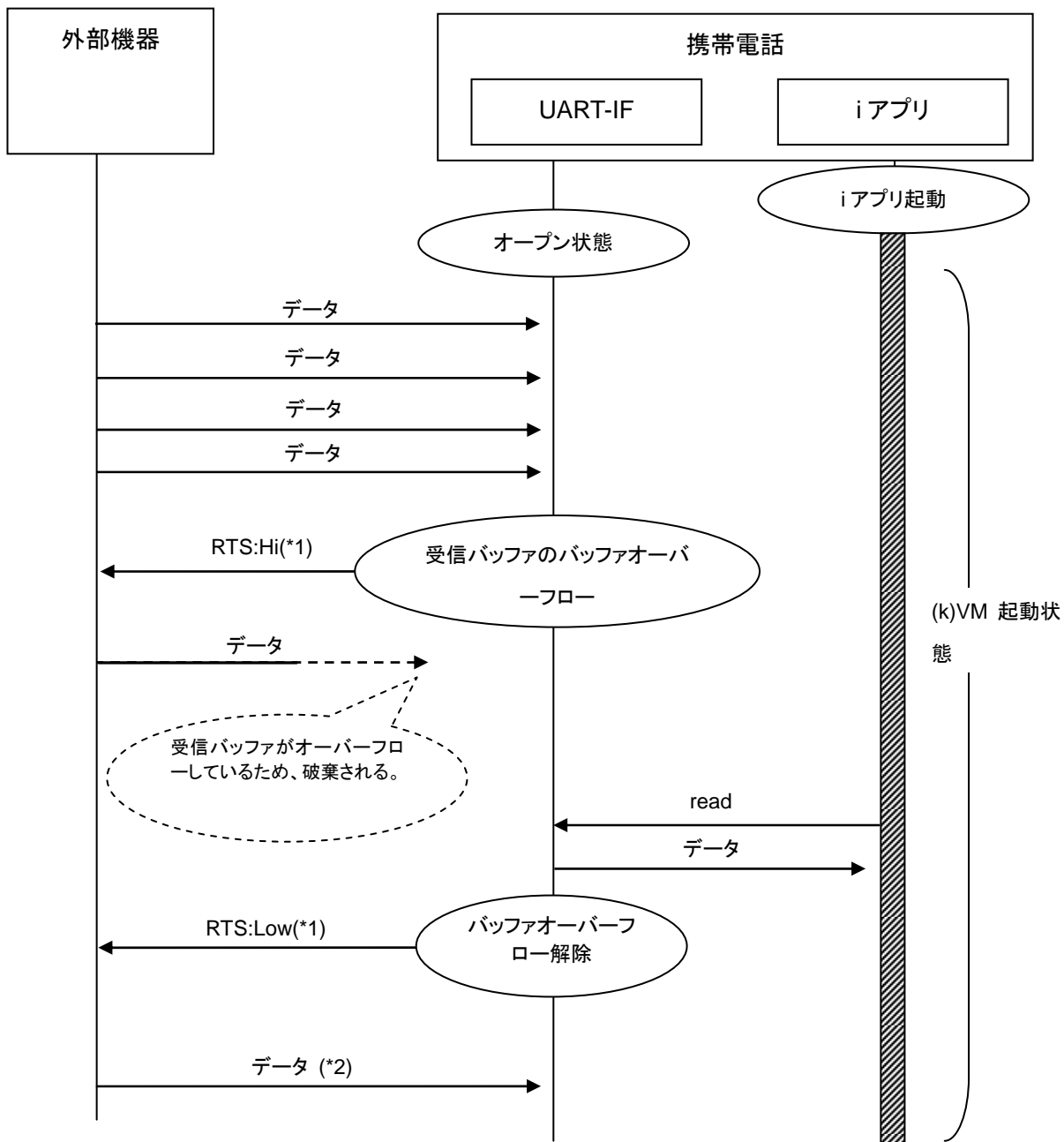
7.1.5 i アプリの終了動作

UART オープン状態でi アプリを終了した場合のシーケンスを以下に示す。



7.1.6 バッファオーバーフロー動作

バッファオーバーフローした場合のシーケンスを以下に示す。

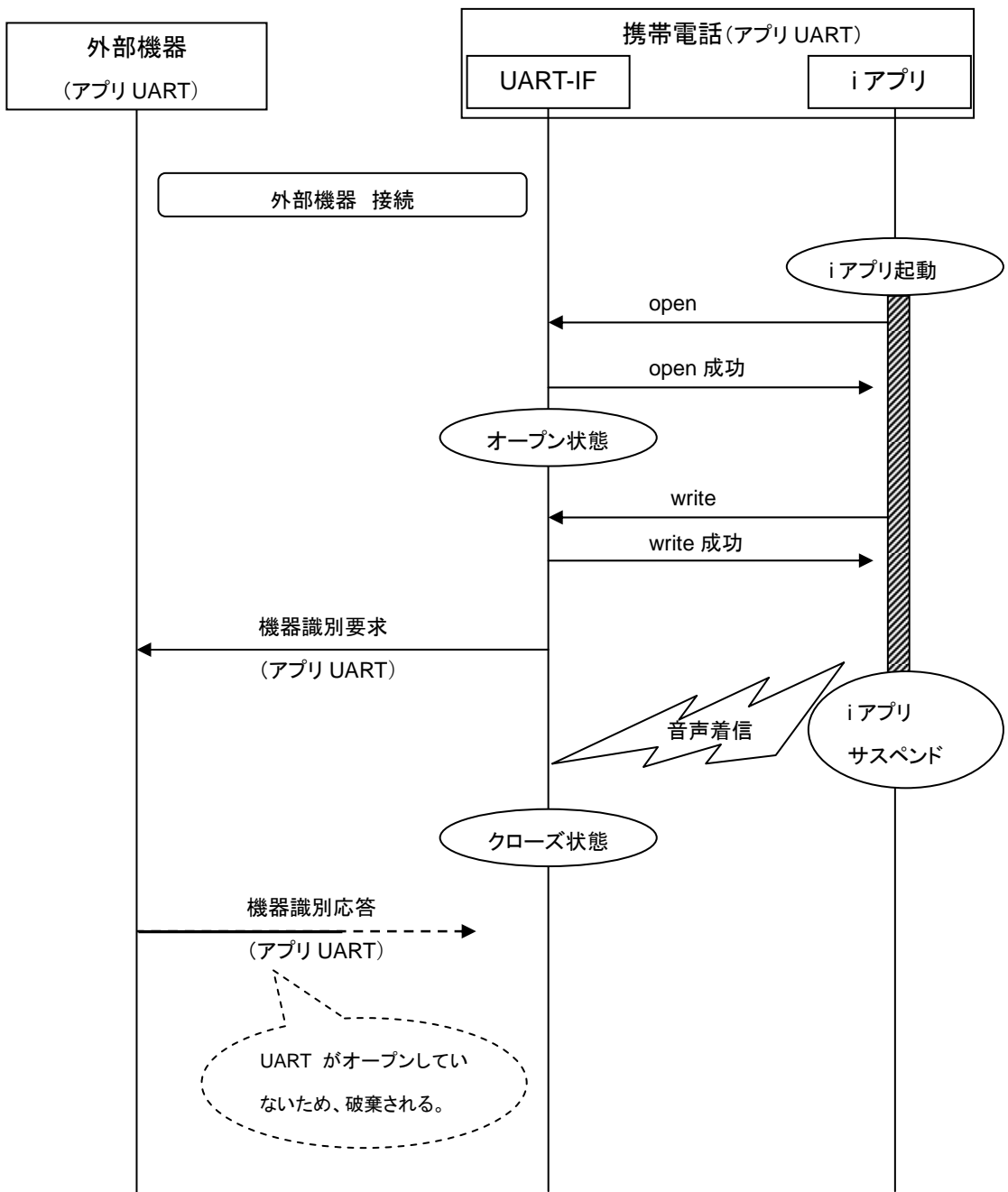


*1: フロー制御を行う場合のみ。

*2: 外部機器のデータが受信バッファに格納される。ただし、受信バッファの空き容量を超えた場合には破棄される。

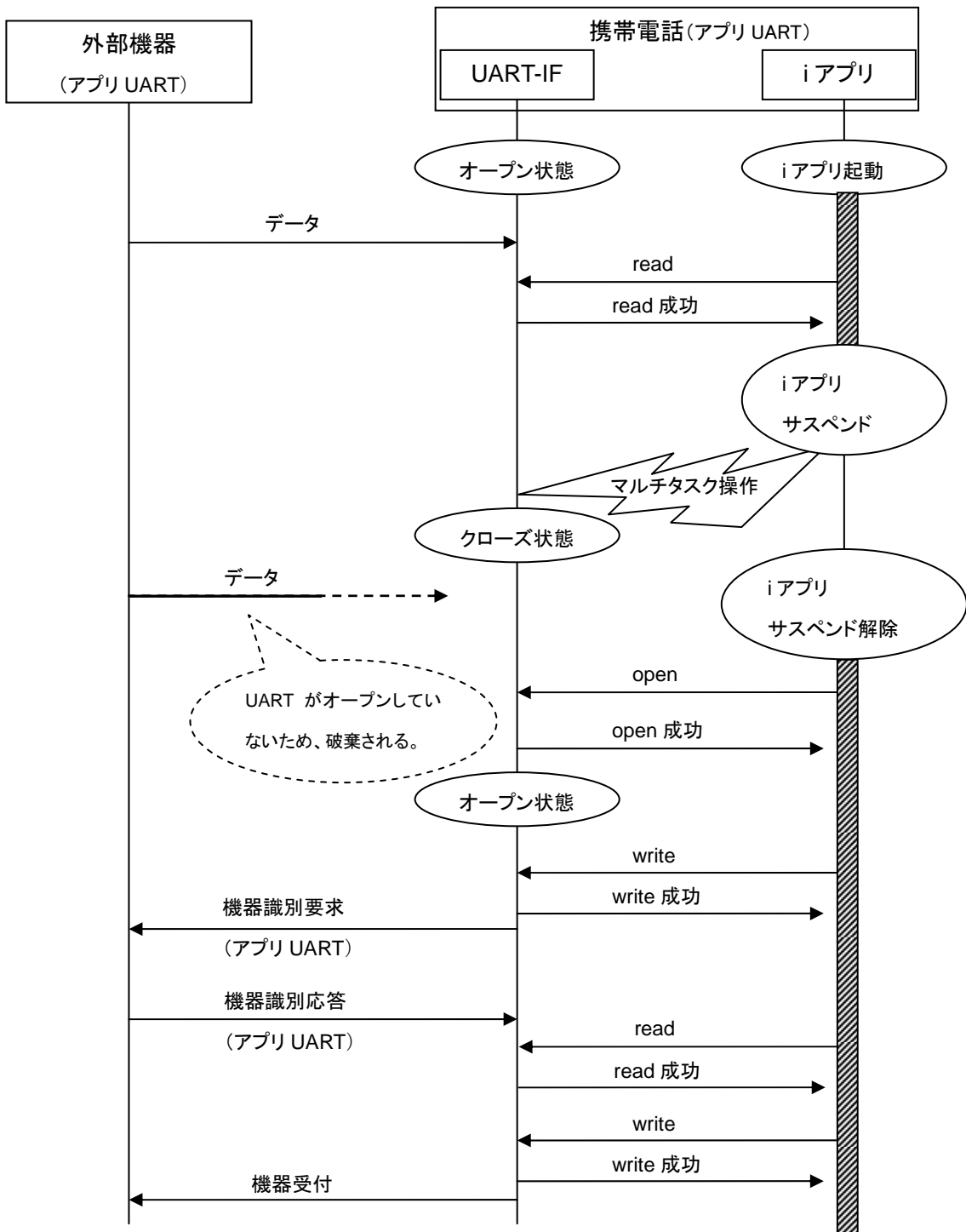
7.1.7 機器識別要求後のiアプリ サスペンド動作

UART がオープンし、機器識別要求(アプリ UART)が送信されたときに、音声着信によりiアプリがサスペンドした場合のシーケンスを以下に示す。



7.1.8 UART クローズ後の UART 再オープン動作

マルチタスクにより UART をクローズした後に UART を再オープンした場合のシーケンスを以下に示す。



8 i アプリから利用する関数

UART 接続によるシリアル通信を行うには J2ME CLDC の Generic Connection Framework を使用する。代表的なメソッドを以下に示す。詳細は J2ME CLDC [3] を参照すること。

Connector クラス

open()
openDataInputStream(String)
openDataOutputStream(String)
openInputStream(String)
openOutputStream(String)

InputConnection インタフェース

openDataInputStream()
openInputStream()

OutputConnection インタフェース

openDataOutputStream()
openOutputStream()

InputStream クラス

available()
close()
read()
read(byte[])
read(byte[], int, int)
reset()
skip(long)

OutputStream クラス

close()
flush()
write(byte[])
write(byte[], int, int)
write(int)

	記載箇所	種別	変更内容	変更前
1.0			初版	
1.1	2	追加	「[2]FOMA USB インタフェースを利用するための技術参考資料」に変更。	
1.1	2	削除	「FOMA サービスを利用するための技術参考資料」を削除。	
1.1	4.5.2	削除	保護回路についての記載を削除。	「なお、保護回路についての記述は省略する。」

ご注意

- (1) 本資料の内容の一部または全部を無断で転載及び複製することは禁止されています。
- (2) 本資料に記載された動作等は、装置の機能追加などにより追加・変更されることがあります。
- (3) 本資料は予告なく改版されることがあります。

(不許複製・禁転載)

FOMA UARTインタフェースを利用するための

技術参考資料

平成20年1月25日 第1.1版 発行

編集・発行

株式会社エヌ・ティ・ティ・ドコモ

移動機開発部

〒239-8536 神奈川県横須賀市光の丘3-5 R&D センタ
