

SONY®

FeliCa

SDK for NFC

ユーザーズマニュアル

NFC アクセスライブラリ編 (Lite)

Contactless IC Card
Software Development Kit

はじめに

本文書は、SDK for NFC に含まれる以下のライブラリについて説明しています。

- NFC アクセスライブラリ

- FeliCa は、ソニー株式会社が開発した非接触 IC カードの技術方式です。
- FeliCa は、ソニー株式会社の登録商標です。
- その他、本文中の会社名や商品名は、該当する各社の商標または登録商標です。
- 本文書の全部または一部の複写、複製および第三者への配布を禁止します。
- 本文書の内容は予告なく変更することがあります。
- 本文書を参照することによって生じた損害について、ソニー株式会社は一切の責任を負いません。

(このページは白紙です。)

目次

1. NFC アクセスライブラリ	1
1.1. API 一覧	1
2. NFC アクセスライブラリ API	2
2.1. API 仕様	2
2.1.1. felicalib_nfc_initialize	2
2.1.2. felicalib_nfc_uninitialize()	3
2.1.3. felicalib_nfc_open()	3
2.1.4. felicalib_nfc_close()	4
2.1.5. felicalib_nfc_start_poll_mode()	5
2.1.6. felicalib_nfc_stop_poll_mode ()	6
2.1.7. felicalib_nfc_start_dev_access()	6
2.1.8. felicalib_nfc_poll_and_start_dev_access()	7
2.1.9. felicalib_nfc_stop_dev_access()	8
2.1.10. felicalib_nfc_select_device()	8
2.1.11. felicalib_nfc_deselect_device()	9
2.1.12. felicalib_nfc_thru()	9
2.1.13. felicalib_nfc_set_timeout()	11
2.1.14. felicalib_nfc_get_timeout()	11
2.1.15. felicalib_nfc_set_poll_callback_parameters()	12
2.1.16. felicalib_nfc_get_last_error()	12
2.1.17. felicalib_nfc_start_logging()	13
2.1.18. felicalib_nfc_stop_logging()	13
2.2. コールバック関数仕様	14
2.2.1. デバイス捕捉時のコールバック	14
2.2.2. デバイス使用権獲得／獲得待ち時のコールバック	15
2.2.3. エラー発生時のコールバック	16
2.3. エラーコード	17
2.3.1. エラーコード種別	17
3. 外部データについて	20
3.1. 外部データ	20
3.1.1. レジストリ	20

(このページは白紙です。)

1. NFC アクセスライブラリ

1.1. API 一覧

■NFCアクセスライブラリのAPI

関数名	説明
felicalib_nfc_initialize()	ライブラリ初期化
felicalib_nfc_uninitialize()	ライブラリ終了化
felicalib_nfc_open()	リーダー/ライタのオープン
felicalib_nfc_close()	リーダー/ライタのクローズ
felicalib_nfc_start_poll_mode()	デバイス捕捉処理の開始
felicalib_nfc_stop_poll_mode()	デバイス捕捉処理の終了
felicalib_nfc_start_dev_access()	デバイス使用権の獲得
felicalib_nfc_poll_and_start_dev_access()	デバイス捕捉およびアクセス権の取得
felicalib_nfc_stop_dev_access()	デバイス使用権の解放
felicalib_nfc_select_device()	デバイス SELECT
felicalib_nfc_deselect_device()	デバイス DESELECT
felicalib_nfc_thru()	デバイス(カード)コマンド発行
felicalib_nfc_set_timeout()	タイムアウト時間の設定
felicalib_nfc_get_timeout()	タイムアウト時間の取得
felicalib_nfc_set_poll_callback_parameter()	デバイス捕捉通知情報のセット
felicalib_nfc_get_last_error()	エラー情報の取得
nelicalib_nfc_start_logging	ロギング開始
nelicalib_nfc_stop_logging	ロギング停止

2. NFC アクセスライブラリ API

2. 1. API 仕様

NFC アクセスライブラリが提供する各関数の仕様の詳細を説明します。

2. 1. 1. felicalib_nfc_initialize

```
bool felicalib_nfc_initialize();
```

NFC アクセスライブラリを初期化します。
NFC アクセスライブラリの使用に際し、本関数の実行が必須です。

<ご注意>

FeliCa ライブラリと NFC アクセスライブラリを同一プロセスで使用することはできません。

Parameters:

なし

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.2. felicalib_nfc_uninitialize()

```
bool felicalib_nfc_uninitialize();
```

NFC アクセスライブラリの終了処理を実行します。

NFC アクセスライブラリの使用終了時に実行必須です。

本関数を実行せずに、FeliCa ライブラリあるいは NFC アクセスライブラリを使用した場合、動作は保証されません。

Parameters:

なし

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.3. felicalib_nfc_open()

```
bool felicalib_nfc_open(  
    char *port_name  
);
```

FeliCa ポートをオープンし、使用可能状態にします。

ポート名書式は、以下のとおりです。

USB+n (64 >= n >= 0)

USB でのポート番号

ポートのオープン後、デフォルトタイムアウト値 (6000ms) を設定します。

Parameters:

Port_name [IN/OUT] オープン対象ポート名

空文字の場合は自動オープン(レジストリからポート名取得)。

オープンしたポート名を返します。

NULL の場合は OUT がありません。

NULL でない場合は、16 バイト以上の領域を確保してください。

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.4. felicalib_nfc_close()

```
bool felicalib_nfc_close( );
```

NFC アクセスライブラリの終了処理を実行します。

NFC アクセスライブラリの使用終了時に実行必須です。

本関数を実行せずに、FeliCa ライブラリあるいは NFC アクセスライブラリを使用した場合、動作は保証されません。

カード捕捉を停止していない場合は、自動停止します。

Parameters:

なし

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.5. felicalib_nfc_start_poll_mode()

```
bool felicalib_nfc_start_poll_mode(  
    unsigned long target_device  
);
```

指定デバイスの捕捉処理を開始します（ライブラリ内部で一定間隔*の捕捉を試みます）。
デバイスを捕捉時およびデバイス使用権を獲得した際は、Windows メッセージにてその旨を通知します。
また、エラー発生時も Window メッセージにて通知します。
本関数の実行前に、felicalib_nfc_set_poll_callback_parameter で捕捉通知情報を指定してください。

* 600ms

Parameters:

target_device [IN] デバイス情報

target_device に指定可能なデバイス種別（値）は、以下のとおりです。
捕捉対象デバイスは複数指定できます（論理和で指定）。

デバイス種別 (Define)	値	説明
DEVICE_TYPE_NFC_14443A_18092_106K	0x00000001	ISO/IEC 14443 TypeA 106Kbps ISO/IEC 18092 106Kbps ※ ISO/IEC 14443-3 デバイス (Mifare Standard, Ultra Light) も含む
DEVICE_TYPE_NFC_18092_212K	0x00000002	ISO/IEC 18092 212Kbps
DEVICE_TYPE_NFC_18092_424K	0x00000004	ISO/IEC 18092 424Kbps
DEVICE_TYPE_NFC_14443B_106K	0x00000008	ISO/IEC 14443-4 TypeB 106Kbps

Return values:

true : 成功
false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.6. felicalib_nfc_stop_poll_mode()

```
bool felicalib_nfc_stop_poll_mode();
```

デバイス捕捉処理を停止します。
NFC アクセスライブラリをアンロードする前に、必ず本関数を実行してください。
本関数を実行せずにライブラリをアンロードした場合、アプリケーションの動作は保証されません。

Parameters:

なし

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.7. felicalib_nfc_start_dev_access()

```
bool felicalib_nfc_start_dev_access(  
    unsigned long target_number  
);
```

捕捉デバイスの使用を宣言し、デバイス使用权を獲得します。
デバイス使用权の獲得に成功した場合、デバイスは SELECT 状態になります。捕捉デバイス識別子は、デバイス捕捉時に NFC アクセスライブラリより Windows メッセージで通知されます。

<ご注意>

本関数では、デバイスにコマンドを送信しません。

Parameters:

target_number [IN] 捕捉デバイス識別子

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.8. felicalib_nfc_poll_and_start_dev_access()

```
bool felicalib_nfc_poll_and_start_dev_access(  
    unsigned long target_device,  
    unsigned long timeout,  
    DEVICE_INFO** device_info  
);
```

デバイス捕捉およびアクセス権の取得を行います。

felicalib_nfc_start_poll_mode() を呼び出し後、felicalib_nfc_stop_poll_mode() を呼び出すまでの間は、本関数を実行することはできません。

本関数実行中、別スレッドなどから felicalib_nfc_start_dev_access() を呼び出した場合、本関数終了まで実行待機します。

本関数実行中、別スレッドなどから felicalib_nfc_stop_dev_access() を呼び出した場合、本関数終了まで実行待機します。

Parameters:

target_device [IN] デバイス種別
「2.1.5 felicalib_nfc_start_poll_mode()」で指定するものと同じです。
timeout [IN] タイムアウト時間 (ミリ秒)
device_info [OUT] デバイス情報
「2.2.1 デバイス捕捉時のコールバック」で得られる情報と同じです。

Return values:

true : 成功
false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.9. felicalib_nfc_stop_dev_access()

```
bool felicalib_nfc_stop_dev_access(  
    unsigned long stop_mode  
);
```

デバイス使用権を解放します。

本関数はデバイスにコマンドを送信しません。

stop_mode に、仕様外の値を指定した場合は、エラーとなります (FELICA_NFC_E_INVALID_ARGUMENT)。

Parameters:

stop_mode [IN] 停止動作種別 (下記表参照)

停止動作種別	値	説明
RE_NOTIFICATION_SAME_DEVICE	0x00000000	同一デバイスを捕捉通知対象とする
NOT_RE_NOTIFICATION_SAME_DEVICE	0x00000001	同一デバイスを捕捉通知対象としない

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1 で定義されているエラーコードの数値

2.1.10. felicalib_nfc_select_device()

```
bool felicalib_nfc_select_device();
```

デバイスを SELECT します。

Parameters:

なし

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1 で定義されているエラーコードの数値

2.1.11. felicalib_nfc_deselect_device()

```
bool felicalib_nfc_deselect_device();
```

SELECT 中のデバイスを DESELECT します。

Parameters:

なし

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.12. felicalib_nfc_thru()

```
bool felicalib_nfc_thru(  
    const unsigned char *command_packet_data,  
    unsigned short command_packet_data_length,  
    unsigned char *response_packet_data,  
    unsigned short *response_packet_data_length  
);
```

デバイス（カード）コマンドを発行し、レスポンスデータを返却します。
デバイス（カード）種別ごとの送受信データは下記のとおりです。

1. 送信時

- 1) ISO/IEC 18092 212K、424K (FeliCa)
カードコマンドのペイロードデータ (LEN+Command)
- 2) ISO/IEC 14443-4 TypeA、TypeB
カードコマンドの Information Field^{※1}
- 3) MIFARE® Ultralight
カードコマンドのペイロードデータ
- 4) MIFARE® Classic (MIFARE® Standard 1K/4K)
カードコマンドのペイロードデータ (AuthenticationA, B を除く^{※2})
command_packet_data に渡すパケットデータを暗号化する必要はありません。
パケットデータの暗号化は、リーダー/ライターが自動的に行います。

1. 受信時

- 1) ISO/IEC 18092 212K、424K (FeliCa)
カードレスポンスのペイロードデータ (LEN+Command)
- 2) ISO/IEC 14443-4 TypeA、TypeB
カードレスポンスの Information Field^{※1}
- 3) MIFARE® Ultralight
カードレスポンスのペイロードデータ
- 4) MIFARE® Classic (MIFARE® Standard 1K/4K)

カードレスポンスのペイロードデータ (AuthenticationA, B を除く※³)
command_packet_data に渡すパケットデータを復号化する必要はありません。
パケットデータの復号化は、リーダー/ライターが自動的行います。

※1

ISO/IEC14443-4 TypeB カードで、Maximum Frame Size Code in ATQB の値が 5 を超える場合、64 バイト (CID を使用する場合は 63 バイト) を超えるデータを指定できません。

※2

MIFARE® Classic に AuthenticationA または B を送信する場合は、カードコマンドのペイロードデータではなく、下記コマンドを指定します(括弧内はバイト数)。

[コマンドコード(1)] + [アドレス(1)] + [鍵(6)] + [UID(4)]

※ コマンドコードは、カードコマンドと同じものを指定してください。

AuthenticationA または B が成功すると、以降、リーダー/ライターとカード間では暗号化(MIFARE® CRYPTO 1)した RF 通信が行われます。

※3

AuthenticationA または B の場合、response_packet_data はありません。

AuthenticationA または B の成否が関数の戻り値として返却されます。

<ご注意>

カードコマンドについては、カード会社にお問い合わせください。

Parameters:

command_packet_data	[IN] デバイスパケットデータ
command_packet_data_length	[IN] command_packet_data データ(バイト)
response_packet_data	[IN/OUT] レスポンスデータ
	262 バイト以上の領域を確保してください
response_packet_data_length	[OUT] レスポンスデータ長

Return values:

true : 成功
false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.13. felicalib_nfc_set_timeout()

```
bool felicalib_nfc_set_timeout(  
    unsigned long timeout  
);
```

デバイスドライバとリーダー/ライタ間のタイムアウト値を設定します。
デフォルト値は 6000ms です。

Parameters:

time_out [IN] タイムアウト値 (ミリ秒)

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.14. felicalib_nfc_get_timeout()

```
bool felicalib_nfc_get_timeout(  
    unsigned long *timeout  
);
```

現在のタイムアウト設定値を取得します。

Parameters:

time_out [OUT] タイムアウト値 (ミリ秒)

Return values:

true : 成功

false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.15. felicalib_nfc_set_poll_callback_parameters()

```
bool felicalib_nfc_set_poll_callback_parameters(
    HWND handle,
    const unsigned char *msg_str_of_find,
    const unsigned char *msg_str_of_enable,
);
```

デバイス捕捉通知情報をセットします。

本関数実行後、felicalib_nfc_start_poll_mode 実行により、デバイス捕捉を開始します。

デバイスを捕捉時およびデバイス使用権を獲得した際には、Window メッセージにてその旨を通知します。

通知 Windows メッセージは、msg_str_find・msg_str_enable から RegisterWindowMessage にて作成します。

Parameters:

Handle [IN] ウィンドウハンドル
msg_str_of_find [IN] デバイス捕捉時の Window メッセージ規定用文字列
msg_str_of_enable [IN] デバイス使用権獲得時の Window メッセージ規定用文字列

Return values:

true : 成功
false : 失敗

Error Types:

表 2-1 で定義されているエラーコードの数値

2.1.16. felicalib_nfc_get_last_error()

```
bool felicalib_nfc_get_last_error(
    unsigned long * error_info,
);
```

直近に実行した関数のエラー情報を取得します。

ドライバエラーが発生している場合、NFC アクセスライブラリエラーコードとドライバエラーコードが error_info に順番に格納されます。

<ご注意>

関数の実行結果が true の場合、エラー情報は不定値となります。

Parameters:

error_info [IN] エラー情報
sizeof(unsigned long) * 2 バイト

Return values:

true : 成功
false : 失敗

Error Types:

表 2-1 で定義されているエラーコードの数値

2.1.17. felicalib_nfc_start_logging()

```
bool felicalib_nfc_start_logging(  
    const char * filename,  
);
```

動作ログの記録を開始します。

ログは指定ファイルに記録されます。

ログファイルは、共有モードでオープンします。

ログ行数が 65535 行を超えた場合、ログファイルを別名(filename + "_old")で保存し、新たにファイルを作成します。ファイル名(filename + "_old")のファイルがすでに存在していた場合、当該ファイルを削除します。削除に失敗した場合は、別名保存を中止し、そのままロギング処理を継続します。

ログファイルに出力する情報は、以下のとおりです。

- ・呼び出し公開関数名(引数データ含む)
- ・公開関数の戻り値(引数データ含む)
- ・カード捕捉またはエラー発生イベント情報

Parameters:

filename [IN] ログファイル名(フルパス)
パス省略不可。
最大 260 バイト(終端 NULL 含む)

Return values:

true : 成功
false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.1.18. felicalib_nfc_stop_logging()

```
bool felicalib_nfc_stop_logging();
```

動作ログの記録を停止します。

Parameters:

なし

Return values:

true : 成功
false : 失敗

Error Types:

表 2-1で定義されているエラーコードの数値

2.2. コールバック関数仕様

コールバック関数の仕様の詳細を説明します。

2.2.1. デバイス捕捉時のコールバック

デバイス捕捉時のコールバック (Window メッセージによる通知) 仕様について説明します。

例として、WndProc() でメッセージを受信する場合を用います。

```
LRESULT CALLBACK WndProc(  
    HWND hWnd,  
    UINT msg,  
    WPARAM wParam,  
    LPARAM lParam  
);
```

Parameters:

```
hWnd    [IN] Window ハンドル  
msg     [IN] Window メッセージ  
wParam [IN] TRUE(1)  
lParam [IN] デバイス情報
```

1) デバイス情報は、以下のとおりです。

```
typedef struct {  
    unsigned long target_device; // 捕捉デバイス(「2.1.5 felicalib_nfc_start_poll_mode()」参照)  
    unsigned char dev_info[1];  // デバイス情報(data_info_size 分データが格納されます)  
} DEVICE_INFO;
```

2) dev_info に格納されるデータは、以下のとおりです。

- DEVICE_TYPE_NFC_14443A_18092_106K の場合

```
typedef struct {  
    unsigned long target_number; // 捕捉デバイス識別子  
    unsigned short sens_res;  
    unsigned char sel_res;  
    unsigned char NFCID1_size;  
    unsigned char NFCID1[10];  
    unsigned char ATS_size; // ATS データサイズ(バイト). 非 14443-4 の場合は 0.  
    unsigned char ATS[1]; // ATS_size 分データが格納される  
} DEVICE_DATA_NFC_14443A_18092_106K;
```

- DEVICE_TYPE_NFC_14443B_106K の場合

```
typedef struct {  
    unsigned long target_number; // 捕捉デバイス識別子  
    unsigned char ATQB[12];  
    unsigned char ATTRIB_size; // ATTRIB データサイズ(バイト)
```

```
    unsigned char ATTRIB [1];          // ATTRIB_size 分データが格納される
} DEVICE_DATA_NFC_14443B_106K;

・ DEVICE_TYPE_NFC_18092_212K / 424K の場合
typedef struct {
    unsigned char id;                  //識別子 (212K の場合は 0、それ以外の場合は 1)
    unsigned long target_number;       //捕捉デバイス識別子
    unsigned char NFCID2[8];           // IDm
    unsigned char Pad[8];              // PMm
    unsigned char RD_size;             // RD データサイズ(バイト)
    unsigned char RD[1];               // RD_size 分データが格納される
} DEVICE_DATA_NFC_18092_212_424K;
```

<ご注意>

DEVICE_INFO データを、本関数内で解放しないでください。

2.2.2. デバイス使用権獲得／獲得待ち時のコールバック

デバイス使用権獲得／獲得待ち時のコールバック (Window メッセージによる通知) 仕様について説明します。

例として、WndProc() でメッセージを受信する場合を用います。

```
LRESULT CALLBACK WndProc(
    HWND hWnd,
    UINT msg,
    WPARAM wParam,
    LPARAM lParam
);
```

Parameters:

```
hWnd    [IN] Window ハンドル
msg      [IN] Window メッセージ
wParam  [IN] ENABLE_ACCESS_TO_TARGET (1)
          WAITING_ACCESS_TO_TARGET (2)
lParam  [IN] target_number
```

2.2.3. エラー発生時のコールバック

felicalib_nfc_start_poll_mode 発行によるデバイス監視中にエラーが発生した場合のコールバック (Window メッセージによる通知) 仕様について説明します。

例として、WndProc() 関数にてメッセージを受信する場合を用います。

```
LRESULT CALLBACK WndProc(  
    HWND hWnd,  
    UINT msg,  
    WPARAM wParam,  
    LPARAM lParam  
);
```

Parameters:

hWnd [IN] Window ハンドル
 msg [IN] Window メッセージ
 wParam [IN] FALSE(0)
 lParam [IN] エラーコードデータ unsigned long *
 (「2.1.16 felicalib_nfc_get_last_error()」参照)

2.3. エラーコード

2.3.1. エラーコード種別

エラーコード種別は、以下のとおりです。

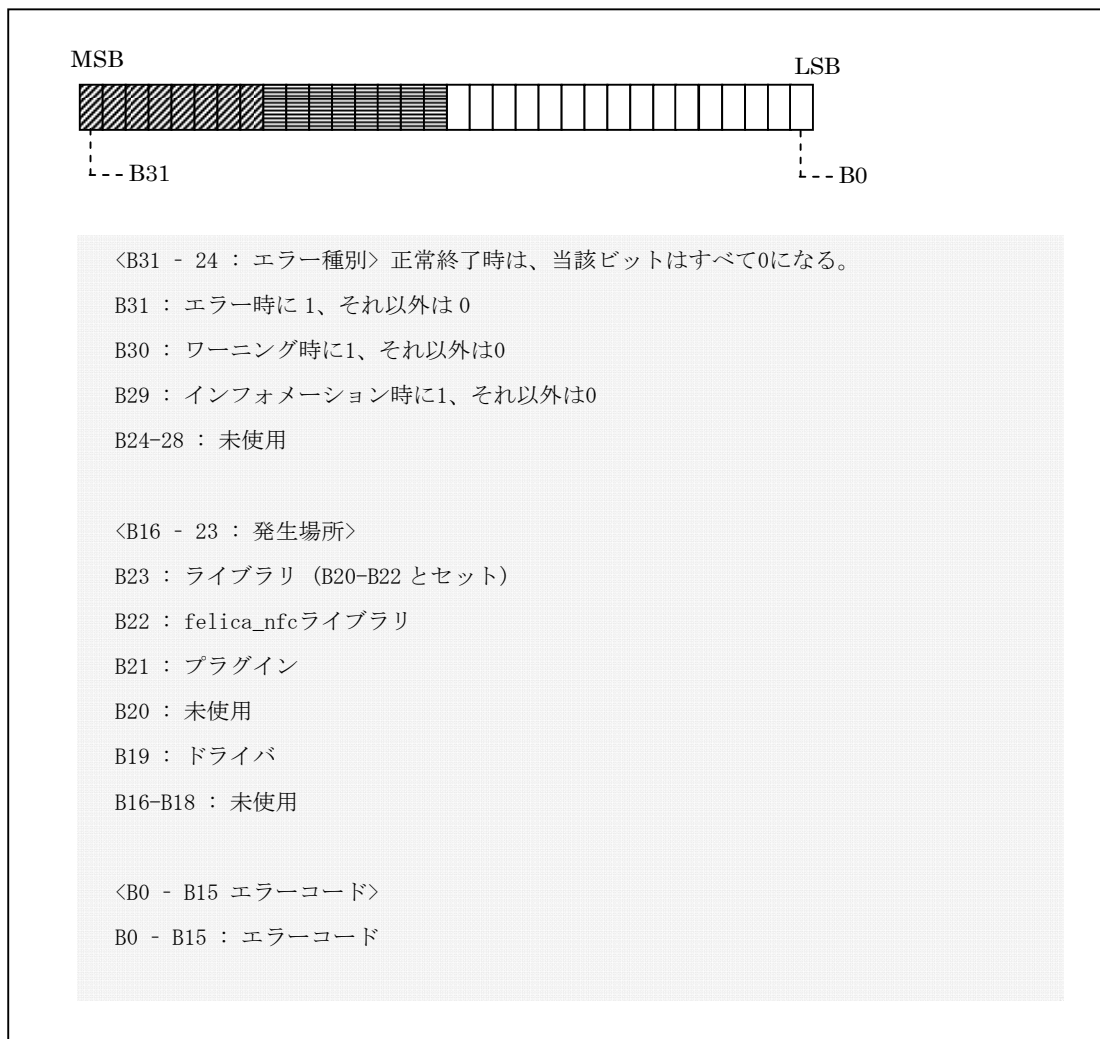


図 2-1: エラーコード種別

表 2-1: エラーコードの定義

名前	値	説明
FELICA_NFC_E_INVALID_ARGUMENT	0x80C00001	引数が異常です。
FELICA_NFC_E_NOT_IMPLEMENT	0x80C00002	実装されていません。
FELICA_NFC_E_INVALID_MODE	0x80C00003	FeliCa ライブラリが動作中です。
FELICA_NFC_E_MEMORY_ALLOCATE	0x80C00004	メモリが確保できません。
FELICA_NFC_E_NOT_INITIALIZE	0x80C00005	ライブラリが初期化されていません。
FELICA_NFC_E_INVALID_PORT_LIST	0x80C00006	ポートリスト名が不正です。
FELICA_NFC_E_INVALID_PORT_NAME	0x80C00007	ポート名が不正です。
FELICA_NFC_E_INVALID_PLUGINS_DIR	0x80C00008	プラグインディレクトリが不正です。
FELICA_NFC_E_INVALID_PLUGIN_NAME	0x80C00009	プラグインファイル名が不正です。
FELICA_NFC_E_INVALID_PLUGIN	0x80C0000a	プラグインファイルが不正です。
FELICA_NFC_E_NO_PLUGIN	0x80C0000b	プラグインファイルが存在しません。
FELICA_NFC_E_CREATE_THREAD	0x80C0000c	スレッドが生成できません。
FELICA_NFC_E_NOT_PORT_OPENED	0x80C0000d	ポートがオープンされていません。
FELICA_NFC_E_NO_CALLBACK_PARAM	0x80C0000e	コールバックパラメータが設定されていません。
FELICA_NFC_E_NOT_CREATE_MESSAGE	0x80C0000f	コールバック用メッセージが生成できません。
FELICA_NFC_E_TIME_OUT	0x80C00010	NFC ライブラリでタイムアウトが発生しました。
FELICA_NFC_E_CALLBACK	0x80C00011	コールバック呼び出しに失敗しました。
FELICA_NFC_E_ALREADY_INITIALIZED	0x80C00012	ライブラリはすでに初期化されています。
FELICA_NFC_E_PORT_OPENED	0x80C00013	すでにポートがオープンされています。
FELICA_NFC_E_DRIVER_STATUS	0x80C00014	ドライバステータスが不正です。
FELICA_NFC_E_INVALID_TARGET_DEVICE_TYPE	0x80C00015	捕捉したターゲットデバイスタイプが不明です。
FELICA_NFC_E_FORBIDDEN_OPERATION	0x80C00016	同期/非同期排他エラーが発生しました。
-	-	-
FELICA_NFC_E_TARGET_DEVICE_NOT_FOUND	0x80C00050	捕捉デバイスが見つかりませんでした。
FELICA_NFC_E_TARGET_DEVICE_TIMEOUT	0x80C00051	捕捉デバイス使用開始要求が時間内にありませんでした。
FELICA_NFC_E_TARGET_DEVICE_ACCESS	0x80C00052	ターゲットデバイスアクセスでのエラーがドライバより通知されました。
FELICA_NFC_E_TARGET_DEVICE	0x80C00053	ターゲットデバイスアクセスでの不明エラーがドライバより通知されました。
-	-	-
FELICA_NFC_E_RW_DEVICE_MODE	0x80C00071	モード移行エラーがドライバ内で発生しています。
FELICA_NFC_E_RW_DEVICE_INITIALIZE	0x80C00072	FeliCa ポートデバイスの初期化処理エラーです。
FELICA_NFC_E_RW_DEVICE_TEMPERATURE_ERROR	0x80C00073	FeliCa ポートデバイスが温度異常状態です。
FELICA_NFC_E_RW_DEVICE_TEMPERATURE_FATAL_ERROR	0x80C00074	FeliCa ポートデバイスが温度異常状態により、使用できない状態です。
FELICA_NFC_E_RW_DEVICE_READER_WRITER_DISCONNECTED	0x80C00075	FeliCa ポートデバイスが切断されました。
FELICA_NFC_E_RW_DEVICE_ERROR	0x80C00076	FeliCa ポートデバイスで不明なエラーが発生しています。
-	-	-
FELICA_NFC_E_REGISTRY	0x80C00101	レジストリアccessに失敗しました。
FELICA_NFC_E_INVALID_DIR	0x80C00102	フォルダが存在しないか不正です。
FELICA_NFC_E_INVALID_FILE	0x80C00103	ファイルが存在しないか不正です。
FELICA_NFC_E_FILE_ACCESS	0x80C00104	ファイルアクセスに失敗しました。
-	-	-
FELICA_NFCP_E_INVALID_ARGUMENT	0x80A00001	プラグインへの引数が異常です。
FELICA_NFCP_E_INITIALIZE_DRIVER	0x80A00002	ドライバの初期化に失敗しました。
FELICA_NFCP_E_MEMORY_ALLOCATE	0x80A00003	メモリが確保できません。
FELICA_NFCP_E_INVALID_PORT_NAME	0x80A00004	ポート名が不正です。
FELICA_NFCP_E_NOT_PORT_OPENED	0x80A00005	ポートがオープンされていません。
FELICA_NFCP_E_OPENED	0x80A00006	すでにポートがオープンされています。

FELICA_NFCP_E_INVALID_RESPONSE	0x80A00007	ドライバからのレスポンスデータが不正です。
FELICA_NFC_E_DRIVER_ERROR	0x80080000	ドライバでエラーが発生しました※ ¹

※¹ ドライバエラービットが 1 の場合、ドライバエラーを取得できます。

NFC アクセスライブラリとドライバエラーの両方がセットされる場合もあります。

表 2-2: ドライバエラーコードの定義

値	説明
0x00000001	リーダー/ライターが抜かれました。
0x00000002	リーダー/ライターが見つかりません。
0x00000003	リーダー/ライターが見つかりません。
0x00000005	リーダー/ライターが抜かれました。
0x00000015	リーダー/ライターがエラーです。 初期化されていない。 温度異常。
0x00000016	リーダー/ライターが抜かれました。
0x00000017	RF 通信エラー、カード状態エラー、リーダー/ライター温度異常、またはカードからの受信データが異常です。
0x00000020	デバイスの最大オープン数を超過しています。
0x00000032	API の発行シーケンスが正しくありません。
0x00000079	アクセス権待ちタイムアウトまたはコマンドタイムアウトが発生しています。
0x0000007A	ドライバインタフェースエラーです。 ※ このエラーが発生した場合、NFC アクセスライブラリに問題がある可能性があります。
0x000000AA	ドライバが処理要求を受け付けられない状態です。
0x000000E9	RF 通信エラー、カード状態エラー、リーダー/ライター温度異常、またはカードからの受信データが異常です。
0x0000045D	USB 通信エラーまたはリーダー/ライターレスポンスパケットが不正です。
0x0000048F	リーダー/ライターが抜かれました。
0x000005AA	システムリソースが不足しています。
0x000006F8	ドライバへの要求パラメータエラーです。

3. 外部データについて

本章では、外部データ仕様を説明します。

3.1. 外部データ

3.1.1. レジストリ

本ソフトウェアが使用するレジストリキーは、以下のとおりです。

OS	レジストリキー
Windows XP Windows Vista Windows 7	HKEY_LOCAL_MACHINE¥SOFTWARE¥Sony Corporation¥FeliCa NFC Library

レジストリ設定項目は、以下のとおりです。

レジストリに値が設定されていない場合、取得できない場合、値が不正な場合は、エラーになります。

レジストリ名	データ型	内容
Plugins_home_directory	文字列 REG_SZ	プラグインの走査対象ルートディレクトリ パス省略不可(フルパスで指定) 最長で MAX_PATH - 1 まで(MAXPATH 以降は無視)
Port_name_list	文字列 REG_SZ	ポート名走査リスト カンマ区切りで指定 (最高 32 個まで。33 個め以降は無視) 最長で 512 バイトまで (514 バイト以降は無視)
Category_name_list	文字列 REG_SZ	プラグインカテゴリ名リスト (最大 256 個まで。257 個め以降は無視) リスト記載の順がカテゴリ番号 (先頭から、0、1、2、、、、255) 最長で 4096 バイトまで(4097 バイト以降は無視)

SDK for NFC ユーザーズマニュアル
NFC アクセスライブラリ編 (Lite)

Version 1.2

2009 年 3 月 初版発行

FeliCa デバイス事業部

2011 年 11 月 改訂

FeliCa 事業部

ソニー株式会社

No. M577-J01-20

© 2009, 2011 Sony Corporation

Printed in Japan