



ピットタッチ・プロ 勤怠パッケージ

サーバ送受信仕様書

Ver2.01

株式会社ビー・ユー・ジー SST

目次

1	この文書の目的	4
2	概要	5
2.1	サーバ通信	5
2.2	再送機能	5
3	サーバ通信	6
3.1	リクエスト送信(勤怠 PKG → Web サーバ)	6
3.1.1	送信パラメータフォーマットのキーワード	7
3.1.2	送信例	12
3.2	レスポンス受信 (Web サーバ → 勤怠 PKG)	14
3.2.1	レスポンスボディ	15
3.2.2	応答例	17
3.3	クロスドメイン問題の回避	19
3.3.1	XMLHttpRequest Level 2 (XHR2) を使う方法	19
3.4	サンプルソース	20
3.4.1	Servlet の場合	20
3.4.2	PHP の場合	23
4	再送機能	25
4.1	サーバへの再送	25
4.1.1	再送待ち追加	25
4.1.2	リクエスト送信 (勤怠 PKG → Web サーバ)	26
4.1.3	レスポンス受信 (Web サーバ → 勤怠 PKG)	26
4.1.4	未送信データとの連動	27
4.1.5	未送信データの消去	27
4.1.6	通信シーケンス	28
5	代理応答	30
6	再生音声	31
6.1	再生音声	31

付録T <u>ユーザーエージェント</u>	32
<u>注意事項・免責事項</u>	33

1 この文書の目的

本書は、ピットタッチ・プロ勤怠パッケージ Version 2.0(以下、勤怠 PKG)を使用して、Web サーバを構築するための技術的仕様についてまとめたドキュメントです。

前提として、Web サーバの知識、FeliCa、MIFARE の基本的知識がある方を対象としています。

勤怠 PKG 製品仕様、基本的な導入手順については製品に付属の取扱説明書を参照ください。

ネットワーク・音声など、勤怠 PKG の設定に関しては「勤怠パッケージ ユーザーマニュアル」を参照ください。

FeliCa や MIFARE、Web サーバの設定に関する質問、サポートは一切できませんので、ご了承ください。

■商標などについて

「ピットタッチ」「ピットタッチ・プロ」は、株式会社ビー・ユー・ジー SST の登録商標です。

「FeliCa」はソニー株式会社が開発した非接触 IC カード技術方式です。

「FeliCa」はソニー株式会社の登録商標です。

「MIFARE」は、NXP セミコンダクターズの登録商標です。

その他の商品名、会社名は、各社の商標または登録商標です。

2 概要

2.1 サーバ通信

勤怠 PKG は HTTP/HTTPS を使用して、FeliCa や MIFARE を検出したことを Web サーバへ通知します。その後、サーバからの応答で勤怠 PKG に音声再生・メッセージ表示・時刻設定などの処理を指示することができます。

但し、設定「オフラインモード」を「利用する」とした場合、サーバ通信は行いません。オフラインモードについては「勤怠パッケージ ユーザーマニュアル」を参照ください。

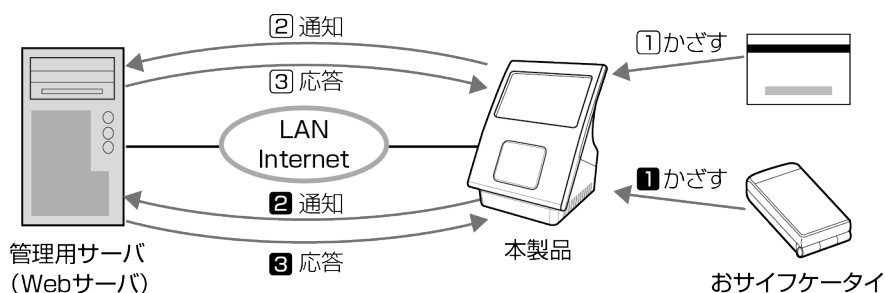


図 1:サーバ通信

2.2 再送機能

勤怠 PKG がサーバ通信を行った結果エラーが発生した場合、再度送信を試みる機能です。但し設定「オフラインモード」を「利用する」とした場合、未送信データが存在していても再送は行いません。

3 サーバ通信

勤怠 PKG に FeliCa や MIFARE のカードやおサイフケータイをかざすと、サーバに HTTP/HTTPS リクエストを用いて各種情報を通知します。

また、このリクエストの応答を用いてサーバから勤怠 PKG に対して音声再生、メッセージ画面表示、時刻修正を指示することができます。

■前提条件

- サーバURL(ホスト名、アプリパス、ポート番号など)他、勤怠PKGの設定が正しく設定されていること。
- WebサーバがHTTP/HTTPSリクエストを受け取れること。
- WebサーバでBasic認証およびDigest 認証を使用していないこと。
- Webサーバでクロスドメイン問題回避の対策をしてあること。(詳細は[3.3 クロスドメイン問題の回避](#)を参照ください)

3.1 リクエスト送信(勤怠 PKG → Web サーバ)

リクエストは勤怠 PKG のサーバ基本設定「送信メソッド」で GET/POST メソッドから選択することができます。また、勤怠 PKG のサーバ基本設定「送信フォーマット」で Web サーバに送信するパラメータ名と値を設定することができます。

送信パラメータフォーマット内に特定のキーワードを指定することで、各種タッチ情報や勤怠 PKG の動作状態を Web サーバに通知することができます。以下の例では[TID]や[CID]など、[]で囲まれたものがキーワードにあたります。

■送信パラメータフォーマット設定例

```
tid=[TID]&type=[TYPE]&cid=[CID]&tim=[TIME]&sts=[STS]&ext=[EXT]&tx=[TX]
&cname=[CNAME]
```

勤怠 PKG ではパラメータ名(上記例での「tid」など)を任意に設定することができます。サーバ側でパラメータ値を取得する際には設定したパラメータ名を指定してください。

3.1.1 送信パラメータフォーマットのキーワード

送信パラメータフォーマット内で指定可能なキーワードには以下のものがあります。

キーワード名	内容
[TID]	端末 ID
[TYPE]	非接触 IC 種別
[CID]	非接触 IC ID
[TIME]	時刻情報
[STS]	ステータス
[EXT]	食事ステータス
[TX]	送信状態
[CNAME]	カード登録名

■ 端末 ID

勤怠 PKG 各機に固有の端末 ID です。勤怠 PKG の設定ページから確認することができます。

キーワード名	[TID]
パラメータ値	8 バイト固定 ASCII 10 進数字

■ 非接触 IC 種別

かざした非接触 IC の種別です。

キーワード名	[TYPE]
パラメータ値	文字列 2 バイト固定 FeliCa : 00 MIFARE standard 1K : A1 MIFARE standard 4K : A2 MIFARE UltraLight : A3

■ 非接触 IC ID

かざした非接触 IC の ID です。FeliCa ID(IDm)または MIFARE UID となります。

【注意】

この ID はカードによって一意にならない場合があります。ご利用されるカードをご確認ください。ID はお客様の責任においてご利用ください。

キーワード名	[CID]
パラメータ値	文字列 20 バイト最大、ASCII 16 進数字

■ 時刻情報

非接触 IC をかざしたときの時刻情報です。勤怠 PKG に設定されている時刻情報を利用します。

キーワード名	[TIME]
パラメータ値	YYYYMMDDhhmmss 14 バイト固定 ASCII 10 進数字 YYYY : 西暦 4 桁(2012, etc.) MM : 月, 2 桁, 01-12 DD : 日, 2 桁, 01-31 hh : 時, 2 桁, 00-23 mm : 分, 2 桁, 00-59 ss : 秒, 2 桁, 00-59

■ステータス

検出時のステータス値を示します。

キーワード名	[STS]
パラメータ値	2バイト固定 ASCII 16進数字 01：ステータス1「出勤」 02：ステータス2「退勤」 03：ステータス3「休憩」/「外出」 04：ステータス4「戻り」/「再入室」 ※ステータス設定「ステータス切り替え」で「使用しない」を選択している場合は値なしとなります

ステータスの値は、図2のように本体のタッチパネル液晶上に表示されているステータスボタンの選択状態とステータス値設定で決まります。設定初期値では「出勤」が「01」、「退勤」が「02」、「休憩」/「外出」が「03」、「戻り」/「再入室」が「04」になります。また、各ステータスボタンに対応するステータス値は設定により変更できます。例えば、「退勤」のステータス値を「04」に設定できます。



図 2:ステータス

■ 食事ステータス

検出時の食事ステータス値を示します。

キーワード名	[EXT]
パラメータ値	2バイト固定 ASCII 16進数字 00：食事ステータス0「注文しない」 01：食事ステータス1「A」 / 「注文する」 02：食事ステータス2「B」 03：食事ステータス3「C」 04：食事ステータス4「D」 ※以下の場合には値なしとなります。 ・ステータス設定「ステータス切り替え」で「使用しない」を選択している場合 ・ステータス設定「食事予約」で「使用しない」を選択している場合 ・ステータスが「出勤」以外の場合

食事ステータスの値は、図3のように本体のタッチパネル液晶上に表示されている食事予約ボタンの選択状態で決まり、「注文しない」が「00」、「A」または「注文する」が「01」、「B」が「02」、「C」が「03」、「D」が「04」になります。



図3:食事ステータス

■送信状態

この通信が非接触 IC を検出した際の通信か、再送による通信かを示します。

キーワード名	[TX]
パラメータ値	1バイト固定 ASCII 10進数字 0：非接触 IC を検出した際の通信(通常送信) 1：サーバへの再送通信

■カード登録名

カードIDに対応する登録名を示します。

キーワード名	[CNAME]
パラメータ値	1~32バイト ASCII(デコード後) ※カード登録名は URL エンコードされていますので、サーバ側でデコードしてください。 ※カード登録名が未登録の場合は値なしとなります。

3.1.2 送信例

例 1) ピットタッチ・ビズ (BFR-420)相当の送信パラメータで、FeliCa カードをかざした場合

■ 勤怠 PKG の設定

- ・送信メソッド : GET
- ・送信フォーマット : tid=[TID]&cid=[CID]&tim=[TIME]&typ=[TYPE]&sts=[STS]
- ・食事予約 : 使用しない
- ・ステータス値(出勤) : 1

■ 検出時の勤怠 PKG の状態

- ・端末 ID : 01234567
- ・時刻情報 : 20120102030405
- ・ステータス : 出勤

■ かざした非接触 IC(FeliCa カード)

FeliCa ID : 0123456789ABCDEF

■ サーバ通信時に勤怠 PKG から送信する内容

tid=01234567&cid=0123456789ABCDEF&tim=20120102030405&typ=00&sts=01

例 2) 勤怠 PKG の設定初期値の送信パラメータで、FeliCa カードをかざした場合

■ 勤怠 PKG の設定

- ・送信メソッド : POST
- ・送信フォーマット :
tid=[TID]&type=[TYPE]&cid=[CID]&tim=[TIME]&sts=[STS]&ext=[EXT]&tx=[TX]&cname=[CNAME]
- ・食事予約 : 使用しない
- ・ステータス値(出勤) : 1

■ 検出時の勤怠 PKG の状態

・ 端末 ID : 01234567

・ 時刻情報 : 20120102030405

・ ステータス : 出勤

■ かざした非接触 IC(FeliCa カード)

FeliCa ID : 0123456789ABCDEF

■ 上記非接触 IC に登録しているカード登録名

カード登録名 : pittouch

■ サーバ通信時に勤怠 PKG から送信する内容

tid=01234567&type=00&cid=0123456789ABCDEF&tim=20120102030405&sts=01&ext=&tx=0&cname=pittouch

3.2 レスポンス受信 (Web サーバ → 勤怠 PKG)

以下の条件でレスポンスを返してください。条件に合致しない場合、勤怠 PKG は音声番号 1004 を再生しエラー表示を行います。

- HTTP StatusCode : 200
- ContentType : text/plain;charset=UTF-8
- [3.2.1 レスポンスボディ](#)に従ってレスポンスボディ部を記述

Web サーバ側でクロスドメイン問題回避の対策を行う必要があります。詳細は [3.3 クロスドメイン問題の回避](#)を参照ください。

条件に合致したサーバ応答を返すことができない場合でも、勤怠 PKG の設定によって代理応答表示させることも可能です([5 代理応答](#))。

3.2.1 レスポンスボディ

サーバから勤怠 PKG に返すレスポンスのボディ部には以下のフォーマットに従って作成してください。

【注意】

レスポンスボディは PlainText で記述してください。HTML で記述した場合はエラーになります。必ず 1 行 1 項目で記述し、1 項目は [データ名]=[内容]のテキストとなるように、レスポンスを返してください。改行コードは、「CRLF(0x0D 0x0A)」です。同一の応答データが複数存在する場合は、初めに検出したものだけを利用の対象とし、他は無視します。

■ リザルトコード[必須]

サーバでの処理結果を表します。この結果は、勤怠 PKG の検出ログに記録されます。

データ名	res
フォーマット	2 バイト固定 ASCII 16 進数字 00 : 成功 それ以外のコードは、サーバでの実装依存

■ 音声番号[任意]

勤怠 PKG で再生する音声を指定します。

詳細は、[6 再生音声](#)を参照ください。

データ名	snd
フォーマット	4 バイト固定 ASCII 10 進数字

■ 表示メッセージ[任意]

勤怠 PKG で表示するメッセージを指定します。指定したメッセージは勤怠 PKG のタッチパネル液晶上に表示されます。メッセージは「確認」ボタンを押すか一定時間(表示設定「メッセージ表示時間」で変更可能)が経過するまで表示され続けます。

データ名	dsp
フォーマット	最大 256 文字までの文字列(文字コードは UTF-8) ※半角英数字は自動的に改行されません。必要に応じて改行タグ を入れてください。 ※メッセージはプロポーションナルフォントにて表示されます。使用する文字によっては 256 文字以内でも表示しきれない場合がありますのでご注意ください。 ※サーバから 256 文字を超えるメッセージを受信した場合は 257 文字以降の内容を無視します(表示しません)。

■ 設定時刻[任意]

勤怠 PKG に設定する時刻を指定します。

【注意】

勤怠 PKG の現在時刻よりも過去の時刻に設定すると、元の時刻を過ぎるまでの間、時刻表示が進まず止まったままとなります。また、ステータス自動変更機能が正常に動作しなくなります。このような場合は電源ボタンを 2 秒以上押しして機能選択画面を表示し、機能選択画面にある [終了] ボタンを押してください。

【注意】

勤怠 PKG はこの応答を受け取ると、即座に本体へ設定します。ネットワーク上の遅延などは考慮されませんので実際の時刻と同じになるとは限りません。また、頻繁に時刻を修正すると過去の時刻が設定され、前述の時刻が止まったままとなる可能性が高くなりますのでご注意ください。時刻をより正確に設定するには NTP サーバによる自動時刻修正をお勧めします。

データ名	tim
フォーマット	YYYYMMDDhhmmss 14 バイト固定 ASCII 10 進数字 YYYY : 西暦 4 桁(2012, etc.) MM : 月, 2 桁, 01-12 DD : 日, 2 桁, 01-31 hh : 時, 2 桁, 00-23 mm : 分, 2 桁, 00-59 ss : 秒, 2 桁, 00-59

3.2.2 応答例

例 1) FeliCa カード情報通知に対する成功応答

- 応答内容
 - ・ 結果：成功
 - ・ 音声：成功音 1

- 応答データ

res=00

snd=1001

例 2) 失敗応答

- 応答内容
 - ・ 結果：失敗(リザルトコード 0x01)
 - ・ 音声：失敗音 2

- 応答データ

res=01

snd=1004

例 3) 成功応答、メッセージ表示画面表示を指示

- 応答内容
 - ・ 結果：成功
 - ・ 音声：成功音 2
 - ・ メッセージ：食事「C」で予約しました。

- 応答データ

res=00

snd=1002

dsp=食事「C」で予約しました。

例 4) 成功応答、時刻設定を指示

■ 応答内容

- ・ 結果 : 成功
- ・ 音声 : 成功音 1
- ・ 設定時刻 : 2012/02/29 10:15:23

■ 応答データ

res=00

snd=1001

tim=20120229101523

【注意】

勤怠 PKG の時刻が過去の時刻に設定されると一定時間画面上の時刻表示が止まったままになることがあります。修正を行うタイミング、修正頻度などにご注意ください。

3.3 クロスドメイン問題の回避

勤怠打刻用のコンテンツセットは本製品内に格納され、そのドメインは「localhost」です。しかしながら、通信を行ってデータを送るサーバは、「localhost 以外」になり、クロスドメインの問題が発生します。勤怠 PKG ではXMLHttpRequest Level 2 (XHR2) を使用してこの問題を回避します。

3.3.1 XMLHttpRequest Level 2 (XHR2) を使う方法

勤怠 PKG はXHR2 をサポートしています。このため、preflight で Web サーバ側で通信を許可すると通信が可能になります。

具体的には、Web サーバ側でのHTTPのOPTIONS リクエスト対応、response ヘッダのAccess-Control 関連の付加などになります。

利用されるサーバのセキュリティポリシーに従ってご対応ください。

参考サイト

- W3C XMLHttpRequest Level 2
<http://www.w3.org/TR/XMLHttpRequest2/>
- W3C Cross-Origin Resource Sharing
<http://www.w3.org/TR/cors/>
- MDN HTTP access control
https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS

3.4 サンプルソース

3.4.1 Servlet の場合

以下に、Servlet で記述したときの簡単なサンプルを記します。

POST で利用した場合です。

```
protected void doOptions(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    response.setHeader("Access-Control-Allow-Origin", "*");
    response.setHeader("Access-Control-Allow-Methods", "GET, POST, OPTIONS");
    response.setHeader("Access-Control-Allow-Headers", "Content-Type");
    response.setContentType("text/plain;charset=utf-8");
    PrintWriter out = new PrintWriter(new OutputStreamWriter(response.getOutputStream()));
    out.print(" ");
    out.close();
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    // 返信を作成する
    // (クロスドメイン問題を回避するためHTTPのOPTIONSリクエスト対応、
    // responseヘッダのAccess-Control関連の付加などを行う)
    response.setContentType("text/plain;charset=utf-8");
    response.setHeader("Access-Control-Allow-Origin", "*");
    response.setHeader("Access-Control-Allow-Methods", "GET, POST, OPTIONS");
    PrintWriter out = new PrintWriter(new OutputStreamWriter(response.getOutputStream(), "UTF-8"));
    StringBuffer responseText = new StringBuffer();
    // 通知された全てのパラメータ名・値を見る
    // Enumeration en = request.getParameterNames();
    // while (en.hasMoreElements())
    // {
    //     String pName = (String)en.nextElement();
    //     System.out.println(pName + ":" + request.getParameter(pName));
    // }
    // 送信パラメータ フォーマットは、以下のように設定
    //
    // tid=[TID]&type=[TYPE]&cid=[CID]&tim=[TIME]&sts=[STS]&ext=[EXT]&tx=[TX]
    // パラメータの取得
    // パラメータ・端末ID
    String tid = request.getParameter("tid");
```

```

// パラメータ・非接触IC種別
String type = request.getParameter("type");
// パラメータ・非接触IC ID
String id = request.getParameter("cid");
// パラメータ・時刻情報
String timeInfo = request.getParameter("tim");
// パラメータ・ステータス
String status = request.getParameter("sts");
// パラメータ・食事ステータス
String extStatus = request.getParameter("ext");
// パラメータ・送信状態
String sendStatus = request.getParameter("tx");
// パラメータ・カード登録名
String cardName = request.getParameter("cname");
// ここで通知された情報を元に処理、返信を作成する
// 以下のように処理を行う
//
// 1) ステータスが出勤(01)で食事ステータスが注文しない(00)以外
//     音声：1002(成功音2)、メッセージ表示あり：予約を承りました。
// 2) ステータスが出勤(01)で食事ステータスが注文しない(00)、またはステータスが退勤
(02)から再入室(04)
//     音声：1019(おつかれさまです)、メッセージ表示なし
// 3) エラー
//     音声：1003(失敗音1)、メッセージ表示あり：エラーが発生しました。
if (status != null && !status.equals(""))
{
    if ((status.equals("01")) &&
        (!extStatus.equals("00")))
    {
        // 1) ステータスが出勤(01)で食事ステータスが注文しない(00)以外
        //     音声：1002(成功音2)、メッセージ表示あり：予約を承りました。
        responseText.append("res=00");
        responseText.append("¥nsnd=1002");
        responseText.append("¥ndsp=予約を承りました。");
    }
    else
    {
        // 2) ステータスが出勤(01)で食事ステータスが注文しない(00)、または
ステータスが退勤(02)から再入室(04)
        //     音声：1019(おつかれさまです)、メッセージ表示なし
        responseText.append("res=00");
        responseText.append("¥nsnd=1019");
    }
}
else
{
    // エラーを返す

```

```
        responseText.append("res=99");
        responseText.append("¥nsnd=1003");
        responseText.append("¥ndsp=エラーが発生しました。");
    }
    out.print(responseText.toString());
    out.close();
}
```

3.4.2 PHPの場合

以下に、PHPで記述したときの簡単なサンプルを記します。

GETで利用した場合です。

```
<?php
// *****
// 注意
// POSTを利用する場合は doGet()、'GET'、$_GET 部分をそれぞれ
// doPost()、'POST'、$_POST に読みかえて下さい
// *****
if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') {
    doOptions();
} else if ($_SERVER['REQUEST_METHOD'] === 'GET') {
    doGet();
}

function doOptions() {
    header("Access-Control-Allow-Origin: *");
    header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
    header("Access-Control-Allow-Headers: *");
    header("Content-type: text/plain;charset=UTF-8");
    echo " ";
}

function doGet() {
    // 送信パラメータ フォーマットは、以下のように設定
    //
    // tid=[TID]&type=[TYPE]&cid=[CID]&tim=[TIME]&sts=[STS]&ext=[EXT]&tx=[TX]
    // パラメータの取得
    // パラメータ・端末ID
    $tid = isset($_GET["tid"]) ? $_GET["tid"] : null;
    // パラメータ・非接触IC種別
    $type = isset($_GET["type"]) ? $_GET["type"] : null;
    // パラメータ・非接触IC ID
    $id = isset($_GET["cid"]) ? $_GET["cid"] : null;
    // パラメータ・時刻情報
    $timeInfo = isset($_GET["tim"]) ? $_GET["tim"] : null;
    // パラメータ・ステータス
    $status = isset($_GET["sts"]) ? $_GET["sts"] : null;
    // パラメータ・食事ステータス
    $extStatus = isset($_GET["ext"]) ? $_GET["ext"] : null;
    // パラメータ・送信状態
    $sendStatus = isset($_GET["tx"]) ? $_GET["tx"] : null;
    // パラメータ・カード登録名
    $cardName = isset($_GET["cname"]) ? $_GET["cname"] : null;
}
```

```

header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
header("Access-Control-Allow-Headers: *");
header("text/plain;charset=UTF-8");
// ここで通知された情報を元に処理、返信を作成する
// 以下のように処理を行う
//
// 1) ステータスが出勤(01)で食事ステータスが注文しない(00)以外
//     音声: 1002(成功音2)、メッセージ表示あり: 予約を承りました。
// 2) ステータスが出勤(01)で食事ステータスが注文しない(00)、またはステータスが退勤
(02)から再入室(04)
//     音声: 1019(おつかれさまです)、メッセージ表示なし
// 3) エラー
//     音声: 1003(失敗音1)、メッセージ表示あり: エラーが発生しました。
if ($status != null)
{
    if (($status === "01") &&
        ($extStatus !== "00"))
    {
        // 1) ステータスが出勤(01)で食事ステータスが注文しない(00)以外
        //     音声: 1002(成功音2)、メッセージ表示あり: 予約を承りました。
        printf("res=00");
        printf("¥r¥nsnd=1002");
        printf("¥r¥ndsp=予約を承りました。");
    }
    else
    {
        // 2) ステータスが出勤(01)で食事ステータスが注文しない(00)、または
ステータスが退勤(02)から再入室(04)
        //     音声: 1019(おつかれさまです)、メッセージ表示なし
        printf("res=00");
        printf("¥r¥nsnd=1019");
    }
}
else
{
    // エラーを返す
    //     音声: 1003(失敗音1)、メッセージ表示あり: エラーが発生しました。
    printf("res=99");
    printf("¥r¥nsnd=1003");
    printf("¥r¥ndsp=エラーが発生しました。");
}
}
?>

```


4 再送機能

勤怠 PKG がサーバ通信を行った結果エラーが発生した場合、再度送信を試みる機能です。設定により機能を利用するかどうかを選択することができます。

4.1 サーバへの再送

勤怠 PKG の設定「再送機能」が「再送する」の場合、非接触 IC をかざした直後のサーバとの通信に失敗しても、のちに同サーバへ再送を行います。

再送待ちが発生するとその情報は未送信データとして内部データベースに保存されます。以後、再送は未送信データの記録に従って行われます。

再送は [4.1.6 通信シーケンス](#) 従い、非接触 IC をかざすことにより発生するサーバ通信([3 サーバ通信](#))とは非同期に行われます。

4.1.1 再送待ち追加

非接触 IC をかざし発生したサーバとの通信に失敗した場合、失敗理由が以下のいずれかであれば再送待ちが発生します。再送待ちが発生すると勤怠 PKG は代理応答表示([5 代理応答](#))を行い、再送待ち情報を未送信データに記録します([4.1.4 未送信データとの連動](#))。

- **サーバ通信エラーが発生**
通信タイムアウト、サーバホスト名解決失敗(DNSエラー)、クロスドメインエラー([3.3 クロスドメイン問題の回避](#))などの理由によりサーバからの応答が正しく得られない場合
- **サーバ応答リプライコードが不正**
サーバから応答を受信したが、HTTPリプライコードが200以外の場合
- **ネットワーク未接続**
非接触ICをかざした際にネットワーク未接続だった場合
- **サーバURLが未設定**
勤怠PKGの設定「サーバURL」が未設定(空)の場合

※以下の場合は再送待ちは発生しません。

- サーバから失敗応答を受信
サーバから応答を正常に受信したが、応答内のリザルトコード([3.2.1 レスポンスボディ](#))が00以外の場合 (res=00以外)
- サーバから未対応フォーマット応答を受信
サーバから応答を受信したが、応答内容が [3.2.1 レスポンスボディ](#) のフォーマットに従わない場合

4.1.2 リクエスト送信 (勤怠 PKG → Web サーバ)

勤怠 PKG は [3.1 リクエスト送信\(勤怠 PKG → Web サーバ\)](#) と同じフォーマットで再送を行います。その際に送信パラメータフォーマットのキーワードに適用される各値は(再送を行う時点の値ではなく)非接触 IC をかざした際の値を用います。

例えば、再送時のキーワード[TIME]に適用される値は(再送を行う時点の時刻ではなく)非接触 IC をかざした際の時刻となります。

ただし、再送時のキーワード[CNAME]に適用される値は再送時に登録されているカード登録名となります。

MEMO

キーワード[TX]([3.1.1 送信パラメータフォーマットのキーワード](#))を利用することにより、勤怠 PKG から送信されたリクエストが再送かどうかをプライマリサーバ側で判別することが可能です。

4.1.3 レスポンス受信 (Web サーバ → 勤怠 PKG)

以下の条件でレスポンスを返してください。

- HTTP StatusCode : 200

【注意】

勤怠 PKG はレスポンスボディを利用しません。また勤怠 PKG は再送通信の結果によらず、結果表示(音声再生やメッセージ表示)を行いません。通常のサーバ通信と同様に音声・メッセージ表示を指示する応答 ([3.2.1 レスポンスボディ](#))を返しても勤怠 PKG は応答内容を無視し、再送は完了となります。

4.1.4 未送信データとの連動

再送待ちが発生すると、対応する未送信データの「結果」に「失敗(再送待ち)」という記述が付加されます。勤怠 PKG はこの「再送待ち未送信データ」の発生時間が古い順にサーバへ再送を 1 件ずつ行います。1 つの再送が完了すると、その未送信データの「結果」が「失敗(再送完了)」に変化します。

※未送信データの内容を確認することはできません(消去することはできます)。

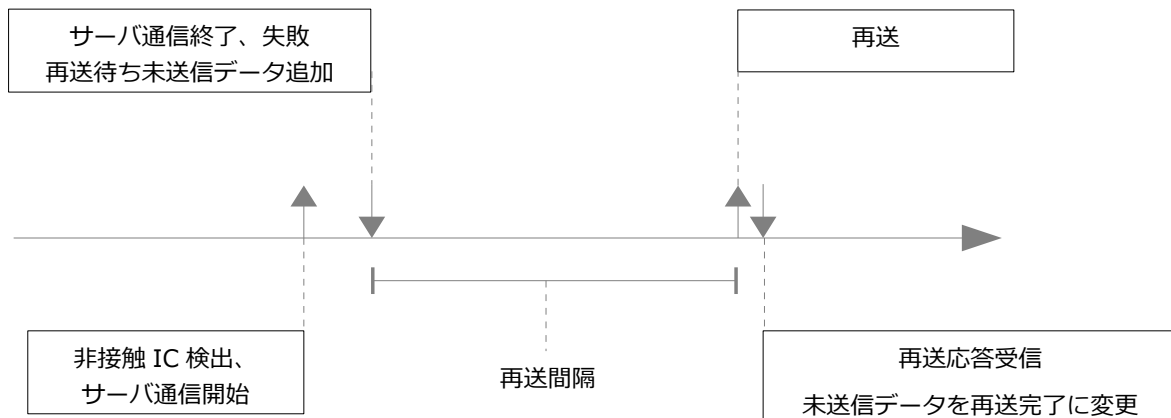


図 4:未送信データの記録

下記に未送信データの記録・変更タイミングの例を示します。

「再送待ち未送信データ」が 1 件以上存在する場合、勤怠 PKG のタッチパネル液晶画面上に「未送信データ」件数を表示します。

勤怠 PKG はカード検出 1 回につき検出ログを 1 件記録します。検出ログを最大件数分記録した状態で新たにカードを検出すると、「再送待ち未送信データ」以外の検出ログがあれば記録時間が古いものから消去し、新たな検出ログを記録します。「再送待ち未送信データ」は自動的に消去されません(ユーザ操作により未送信データが消去される場合があります([4.1.5 未送信データの消去](#)))。「再送待ち未送信データ」が蓄積され、最大件数分記録されると、勤怠 PKG のタッチパネル液晶画面に「未送信データフル画面」が表示されます。

4.1.5 未送信データの消去

未送信データを消去すると再送待ちの情報が消去され、再送が行われなくなります。未送信データは以下の方法で消去されます。ただし、いずれの方法も未送信データを含むすべての検出ログが削除されますので、ご注意ください。

- 基本設定「データベースインデックス」の値(未送信データ格納用データベースのインデックス番号。データベースを初期化したい場合はこの値を変更する)を変更して、コンテンツセットを再起動する
- Web設定ページから設定を初期化する
- オフラインモードでメイン画面を表示する際に表示される「検出ログ削除確認画面」で「削除する」ボタンを押す
なお、設定および設定消去の詳細な手順は「勤怠パッケージ ユーザマニュアル」を参照ください。

4.1.6 通信シーケンス

勤怠 PKG は再送待ちが発生すると、設定「再送間隔」経過後に再送を行います(図 5)。

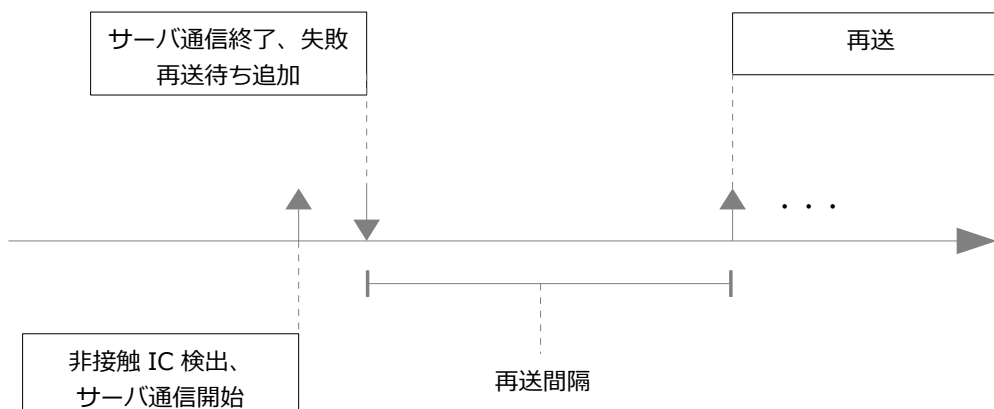


図 5:再送待ち追加、再送実行

再送待ち未送信データが存在する状態でネットワークが接続されると、その直後から再送を行います。よって起動直後にネットワーク接続する状況では、その直後に再送を行います(図 6 再送①)。その後まだ再送待ち未送信データが存在する場合、「再送間隔」経過後に再び再送を行います(図 6 再送②)。

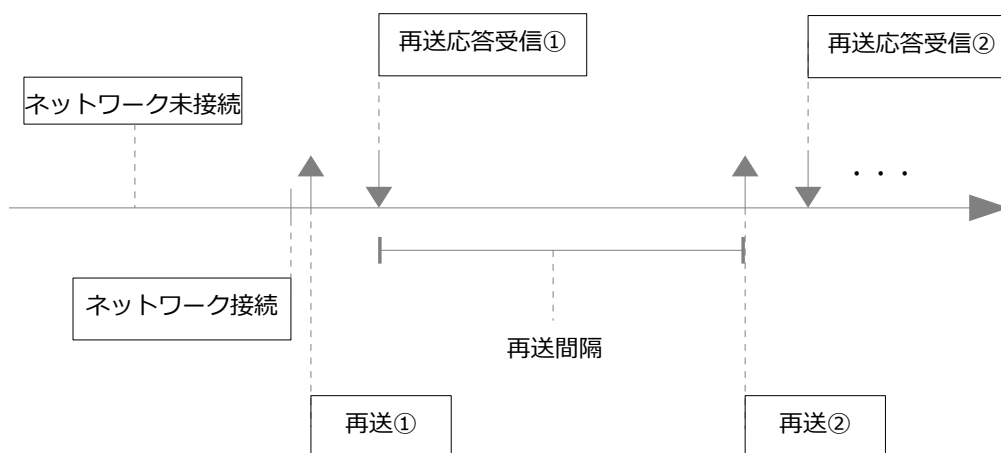


図 6:ネットワーク接続直後から再送実行

「再送待ち未送信データ」が複数存在する場合、発生時間が古い順にサーバへ再送を1件ずつ行います。再送が全て完了し「再送待ち未送信データ」がなくなるか、または未送信データが消去されるまで再送を繰り返します。再送が完了した場合、次の再送待ち未送信データがあればこれに基づいて「再送間隔」経過後に再送を行います。再送が失敗した場合、「再送間隔」経過後に再び同じデータで再送を行います。図7は再送待ち未送信データA、B、C(発生順)を順に再送していく様子を示しています(Aの再送に1度失敗しています)。

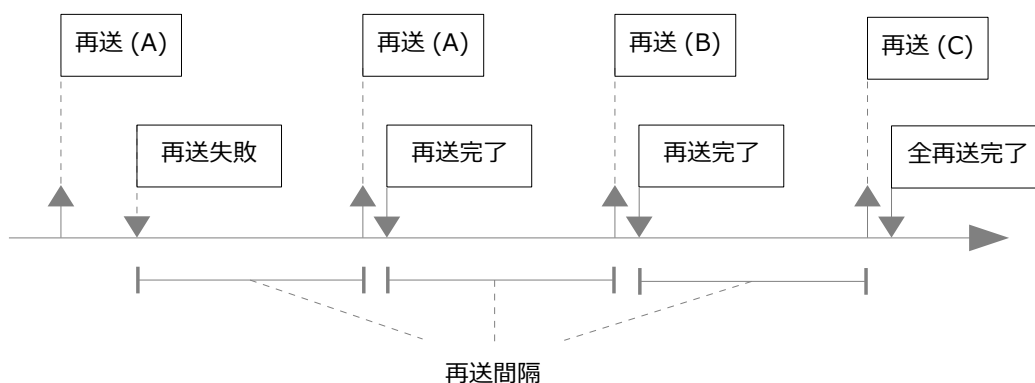


図 7:再送間隔ごとに再送実行

5 代理応答

勤怠 PKG は設定とサーバ通信の結果に応じ、代理応答表示を行います。

- **未対応フォーマット応答受信時の代理応答**

勤怠PKGの設定「未対応フォーマット応答受信」が「代理応答設定に従う」であり、[3.2 レスポンス受信 \(Webサーバ → 勤怠PKG\)](#)に従わないサーバ応答を受信した場合

- **再送待ち発生時の代理応答**

勤怠PKGの設定「再送機能」が「再送する」であり、サーバ通信に失敗した(後に再送を行う)場合

代理応答機能を用いると、勤怠 PKG にレスポンス仕様外のサーバ応答を返してもエラー表示ではなく予め設定しておいた再生音声(ステータス毎に設定可能)を用いた表示をすることができます。これにより既存サーバアプリケーションの応答内容を変更せずに勤怠 PKG を運用することができます。

また、設定「接続タイムアウト」を 0 に設定すると、非接触 IC 検出直後のサーバ通信を行わないようにすることができます。さらに代理応答機能、再送機能を組み合わせることで、ユーザーに対して直ちに打刻完了を通知し、のちほど再送によってサーバ通信を行うことができます。

※設定の詳細な手順は「勤怠パッケージ ユーザマニュアル」を参照ください。

6 再生音声

6.1 再生音声

サーバから勤怠 PKG に対して指示できる再生音声および代理応答機能で設定できる再生音声を示します。

音声番号	主な使用用途/音声内容
1001	成功音 1
1002	成功音 2
1003	失敗音 1
1004	失敗音 2
1008	おはようございます
1009	お帰りなさい
1010	おつかれさまでした
1011	いってらっしゃい
1012	さようなら
1013	登録がありません
1014	出勤です
1015	退勤です
1016	外出です
1017	戻りです
1018	エラーです
1019	おつかれさまです

※代理応答を使用する場合の設定方法は、「勤怠パッケージ ユーザマニュアル」を参照ください。

付録T ユーザーエージェント

勤怠 PKG から Web サーバへアクセスする際に使用するユーザーエージェントは、以下のとおりです。

```
Mozilla/5.0(QtEmbedded; U; Linux; ja-JP; PitTouchPro Build/XXX-nnnnnnnn)  
AppleWebKit/aaa.a (KHTML,like Gecko) Qt/b.b.b Safari/aaa.a
```

XXX : ピットタッチ・プロ ファームウェアバージョン

nnnnnnnn : 本製品の端末 ID

aaa.a : WebKit バージョン、ファームウェアバージョン Ver2.5 では「533.3」

b.b.b : Qt バージョン、ファームウェアバージョン Ver2.5 では「4.7.3」

注意事項・免責事項

- 本書の内容の全部または一部を無断で転載、複製、改変などを行うことは、固くお断りします。
- 本書に記載された仕様、その他の内容については、予告なしに変更することがあります。
- 本書に記載された事項を元にお客様が作成したプログラム・仕様書などについて発生した、いかなる損害も当社は一切の責任を負いません。

2015/09