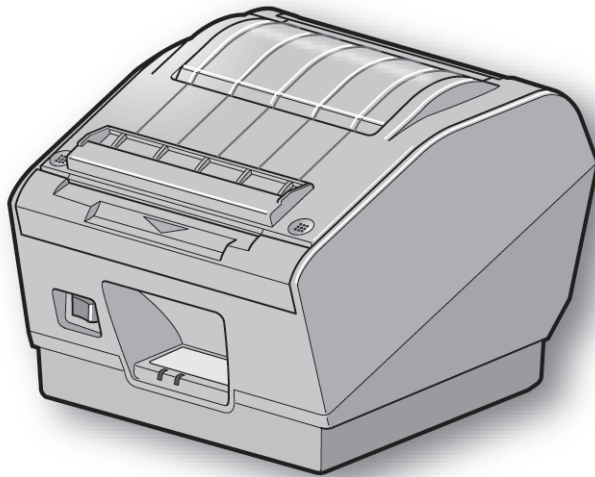


# **Label Printer TSP828L**

## ***Software Development Kit Manual***



本書中に登場する会社名や商品名は、一般に各社の商標または登録商標です。

## 目 次

|                               |   |
|-------------------------------|---|
| 1. ステータスマニタ API.....          | 1 |
| 1.1 GetStarPrinterStatus..... | 3 |
| 2. 改訂履歴 .....                 | 4 |

## 1. ステータスマニタ API

### 概要

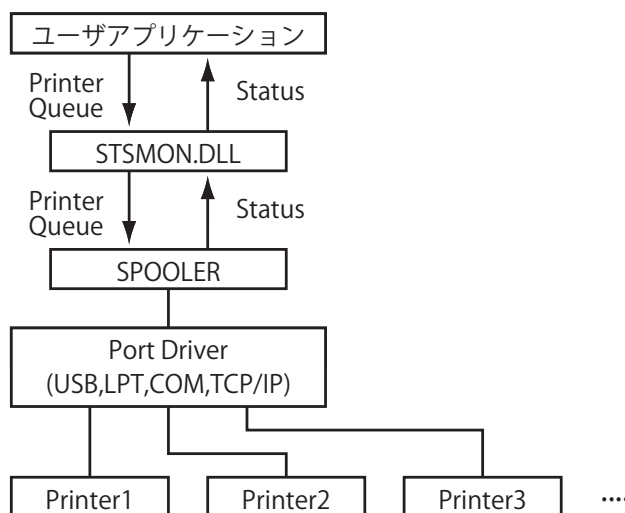
TSP828L ランゲージモニタは、カスタムアプリケーション用に API を提供しています。これらの API を使用することにより、プリンタの状態を取得することができます。

#### ■ プリンタステータス取得の流れ

本機能を利用する場合は、メモリスイッチを以下のように設定してください。

|          |            |          |
|----------|------------|----------|
| シリアル I/F | MSW7-C = 1 | : ASB 有効 |
| パラレル I/F | MSW7-8 = 1 | : NSB 有効 |

ユーザアプリケーションは、STSMON.DLL を介してスプーラと通信を行い、プリンタステータスを取得します。



#### ■ 取得可能なステータス情報

取得できるプリンタステータスは以下の通りです。

- Parsed Status (分析されたステータス)  
各フラグが "TRUE" の場合エラーステータスとなります。
- Raw Status  
64 バイトの分析されていないステータスデータです。

### 関数

| 関数名                  | 処理内容             |
|----------------------|------------------|
| GetStarPrinterStatus | プリンタステータスを取得します。 |

### StarPrinterStatus 構造体

ステータスマニタ API は、StarPrinterStatus\_2 構造体にプリンタの状態を格納します。StarPrinterStatus\_2 構造体は次ページの通りです。

```
typedef struct StarPrinterStatus_2_
{
    // printer status 1
    BOOL coverOpen;                // カバーの状態
    BOOL offline;                  // プリンタ オンライン / オフライン状態
    BOOL compulsionSwitch;         // コンパルジョン SW

    // printer status 2
    BOOL overTemp;                 // 印字ヘッド高温による停止
    BOOL unrecoverableError;       // 復帰不可能エラー
    BOOL cutterError;              // 未使用
    BOOL mechError;                // 未使用
    BOOL headThermistorError;      // 未使用

    // printer status 3
    BOOL receiveBufferOverflow;    // 受信バッファオーバーフロー
    BOOL pageModeCmdError;         // 未使用
    BOOL blackMarkError;           // BM エラー
    BOOL presenterPaperJamError;   // 未使用
    BOOL headUpError;              // 未使用
    BOOL voltageError;             // 未使用

    // printer status 4
    BOOL receiptBlackMarkDetection; // 未使用
    BOOL receiptPaperEmpty;         // 用紙エンド
    BOOL receiptPaperNearEmptyInner; // 未使用
    BOOL receiptPaperNearEmptyOuter; // 未使用

    // printer status 5
    BOOL presenterPaperPresent;     // 未使用
    BOOL peelerPaperPresent;        // はく離センサ用紙検出 あり / なし
    BOOL stackerFull;               // 未使用
    BOOL slipTOF;                   // 未使用
    BOOL slipCOF;                   // 未使用
    BOOL slipBOF;                   // 未使用
    BOOL validationPaperPresent;    // 未使用
    BOOL slipPaperPresent;          // 未使用

    // printer status 6
    BOOL etbAvailable;              // ETB Available
    UCHAR etbCounter;               // ETB counter

    // printer status 7
    UCHAR presenterState;           // 未使用

    // raw
    UINT32 rawLength;               // Raw data Length
    UCHAR raw[63];                  // Raw data
} StarPrinterStatus_2;
```

## 1.1 GetStarPrinterStatus

プリンタステータスを取得します。

```
BOOL GetStarPrinterStatus(
    LPCSTR queueName,           // プリンタキュー名
    StarPrinterStatus_2 * status // StarPrinterStatus_2 構造体へのポインタ
);
```

### パラメータ

#### **queueName**

[in] プリンタキュー名を保持している、NULL で終わる文字列へのポインタを指定します。プリンタキュー名は「プリンタと FAX フォルダ」内のプリンタ名です。

#### **status**

[out] プリンタステータスを格納する StarPrinterStatus 構造体へのポインタを指定します。

### 戻り値

関数が成功すると TRUE、失敗すると FALSE の値が返ります。

### 解説

GetStarPrinterStatus 関数を実行すると、status パラメーターで指定したバッファへプリンタのステータス情報が格納されます。取得したステータス情報の各メンバの値を確認することで、プリンタの状態を判定できます。

また、拡張エラー情報を取得するには、Windows 標準ライブラリの GetLastError 関数を使用します。GetLastError 関数が返すエラーコードは以下の内容を示します。

**ERROR\_NOT\_READY**     プリンタ電源 OFF  
**ERROR\_BROKEN\_PIPE**   プリンタ使用中（印字中など）

### サンプルプログラム

Visual Studio Basic および、Visual Studio C++ で作成したサンプルプログラムを用意してあります。サンプルプログラムは、GetStarPrinterStatus 関数を使用して取得したステータスの状態を表示します。詳しくは Readme\_jp.txt をご参照ください。

## 2. 改訂履歴

[illegible]



**特機事業部**

〒424-0066

静岡県静岡市清水区七ツ新屋 536

電話：054-347-0112 ( 営業直通 )

<http://www.star-m.jp/dl/dl02.htm>