

# VB6.0 ソースコード説明書

## レール軌道波形復元システム

作成日: 2025年10月15日 対象: 20\_復元関係ソースコード 開発環境: Visual Basic 6.0

### 目次

- [1. プロジェクト概要](#)
- [2. 説明（業務未経験者向け）](#)
- [3. ファイル構成](#)
- [4. プロジェクト別詳細](#)
- [5. 主要フォーム説明](#)
- [6. 共通モジュール説明](#)
- [7. データフローと処理ロジック](#)
- [8. 波形復元アルゴリズム](#)
- [9. コード間依存関係](#)
- [10. 注意点と改善ポイント](#)

### プロジェクト概要

このVB6.0プロジェクト群は、鉄道レールの軌道状態を測定した波形データを復元・解析するシステムです。主な機能は以下の通りです：

#### 主要機能

- 軌道波形データの復元処理 - 測定された軌道データから正確な波形を復元
- データ変換・加工 - Oracle、LABOCS、DCP形式間のデータ変換
- 統計解析・計算 - MTT値（軌道変位）、カント、スラック等の計算
- 波形フィルタリング - FFT、ローパスフィルタ等による波形処理
- データベース連携 - Oracle DatabaseやDAO経由でのデータ管理
- レポート生成 - Excel出力、MDTファイル生成

#### 対象システム

- 軌道検測車データ（LABOCS、DCP）
- Oracle軌道データベース
- 軌道管理システム（KCDW）

### 説明（業務未経験者向け）

#### このアプリは何をするの？

このシステムは、電車が安全に走るために、線路の状態をチェックして管理するためのソフトウェアです。

#### 実際に何をしているの？

##### 1. データを集める

- 検測車（けんそくしゃ）という特殊な電車が、線路の上を走りながら、線路のゆがみ具合を測定します

##### 2. データを読み取る

- 検測車が記録したデータ（DCP形式、LABOCS形式など）を、このソフトウェアが読み込みます

### 3. データを分析する

- 線路のどこが、どのくらい曲がっているのか、ゆがんでいるのかを計算します

### 4. わかりやすくする

- 複雑なデータを、グラフや表にして見やすくします
- Excelファイルとして出力できるので、誰でも確認できます

### 5. 必要な場所を見つける

- 特に注意が必要な場所（大きくゆがんでいる場所）を自動的に見つけます

## 具体的にどんな種類のアプリがあるの？

このシステムには6つの異なるアプリ（プログラム）があります：

#### 1. KANA3（カナ3）

- 簡易計算ツール
- 線路のデータを手軽に確認して、簡単な計算をするためのアプリ
- 例：「この区間の線路は、どのくらい傾いているかな？」を素早く調べる

#### 2. KCDW（ケーシーディーダブリュー）

- データ管理システム
- 大量の線路データを保存・管理するためのアプリ
- 例：「過去5年間の線路データをすべて保存して、いつでも検索できるようにする」

#### 3. DCP2S（ディーシーピーツーエス）

- データ変換ツール
- 検測車が記録した生データ（DCP形式）を、別の形式（LABOCS形式）に変換するアプリ
- 例：「カメラで撮った写真（JPG）を、別の形式（PNG）に変換する」のと同じ

#### 4. DCPZW（ディーシーピーゼットダブリュー）

- 座標データ処理ツール
- 線路の縦方向（上下）と横方向（左右）のゆがみを処理するアプリ
- 例：「建物が垂直に立っているか、傾いているかをX軸・Y軸で測定する」のと同じ

#### 5. Ora2Lab2（オラクルツーラボ2）

- データベース変換ツール（通常版）
- 大きなデータベース（Oracle）から線路データを取り出して、使いやすい形式に変換するアプリ
- 例：「図書館の巨大な書庫から、必要な本だけを探し出して、読みやすくコピーする」のと同じ

#### 6. Ora2LaS2（オラクルツーラボエス2）

- データベース変換ツール（簡易版）
- Ora2Lab2の簡易版で、基本的なデータだけを素早く変換するアプリ
- 例：「図書館で本の目次だけを素早くコピーする」のと同じ

## なぜこれが大切なの？

### 安全のため

- 線路がゆがんだまま放置すると、電車が脱線する危険があります
- このシステムで早期発見することで、事故を防ぎます

### 効率的なメンテナンスのため

- すべての線路を手作業でチェックするのは不可能です（何千キロもある）

- このシステムで「本当に修理が必要な場所」だけを見つけて、効率よく修理できます

コスト削減のため

- 問題が小さいうちに見つけて直せば、修理費用も安く済みます
- 大きな事故が起きてからでは、莫大なコストがかかります

どんな人が使うの？

- 鉄道会社の保線担当者：線路の保守・点検をする人
- 技術管理者：線路の状態を分析して、修理計画を立てる人
- 経営層：全体の線路の状態を把握して、予算を決める人

このシステムの「すごいところ」

1. 自動化されている

- 人間が手作業で計算すると何日もかかることを、数分で終わらせます

2. 正確

- 人間の目では見えないわずかなゆがみも、数値として正確に測定できます

3. 履歴管理

- 過去のデータと比較して、「前回より悪くなっているか」を判断できます

4. わかりやすい

- 複雑なデータをグラフや表にして、専門家でなくても理解できるようにします

まとめ：一言で言うと？

「電車の安全を守るために、線路の健康状態をチェックして、『どこを直すべきか』を教えてくれる賢いシステム」です。

ファイル構成

プロジェクト構成（6プロジェクト）

プロジェクト名	説明	メインフォーム	実行ファイル
KANA3	簡易軌道波形パラメータ復元計算	KANA3.frm	KANA3.exe
KCDW	軌道変位データ処理システム	KCDW.frm	KcdW.exe
DCP2S	DCP形式データ変換ツール	DCP2S.frm	DCP2S.exe
DCPZW	DCP座標波形処理	DCPZW.frm	DCPZW.exe
Ora2Lab2	Oracle→LABOCS変換（通常版）	Ora2Lab2.frm	Ora2Lab2.exe
Ora2LaS2	Oracle→LABOCS変換（簡易版）	Ora2LaS2.frm	Ora2LaS2.exe

フォームファイル（.frm）一覧（14ファイル）

ファイル名	行数	イベントハンドラ	所属プロジェクト	主な機能
KANA3.frm	5,820	1	KANA3	メイン計算画面
KCDW.frm	8,467	6	KCDW	軌道変位データ管理

DCP2S.frm	4,365	0	DCP2S	DCP変換メイン
DCP2SA.frm	2,767	-	DCP2S	DCP変換サブ
DCPZW.frm	2,279	-	DCPZW	座標波形処理
Ora2Lab2.frm	9,581	1	Ora2Lab2	Oracle変換（詳細）
Ora2LaS2.frm	5,599	0	Ora2LaS2	Oracle変換（簡易）
KANA3A.frm	276	-	KANA3	サブフォーム A
KANA3B.frm	774	-	KANA3	サブフォーム B
KANA3C.frm	1,507	2	KANA3	サブフォーム C
KANA3D.frm	1,114	1	KANA3	サブフォーム D
KANA3E.frm	1,003	1	KANA3	サブフォーム E
KANA3E.frm	324	-	KANA3	サブフォーム F
KANA3G.frm	717	-	KANA3	MTT検査データコピー

合計: 44,593行

モジュールファイル（.bas）一覧（6ファイル）

ファイル名	行数	関数数	主な機能
CmdLib.bas	33,843	639+	コマンド処理ライブラリ（最重要）
mylib2.bas	13,784	238+	汎用ライブラリ2（拡張版）
KANA3lib.bas	874	13	KANA3専用ライブラリ
KANA3lib2.bas	874	13	KANA3専用ライブラリ2
mylib.bas	571	34	汎用ライブラリ1（基本版）
KcdwKANA3.bas	7	-	KCDW-KANA3連携定数

合計: 49,953行

外部参照ライブラリ

- OLE Automation (stdole2.tlb) - 全プロジェクト
- Microsoft Scripting Runtime (scrrun.dll) - 全プロジェクト
- Microsoft DAO 3.51 (DAO350.DLL) - KCDW
- Microsoft ADO 2.5 - Ora2Lab2
- Common Dialog Control (comdlg32.ocx) - DCP2S, DCPZW, Ora2Lab2
- HgsLACCommon, HgsLZCommon, HgsLAClient, HgsLZClient - Oracle変換系

プロジェクト別詳細

1. KANA3プロジェクト

目的: 簡易軌道波形パラメータ復元計算システム

構成ファイル

- メインフォーム: KANA3.frm
- サブフォーム: KANA3A.frm ~ KANA3G.frm (7個)
- モジュール: myLib.bas, KANA3lib.bas, CmdLib.bas, myLib2.bas

主な機能

- 軌道波形データの読み込みと表示
- MTT値（軌道変位）の計算
- カント、スラック補正
- 波形フィルタリング（FFT処理）
- 結果のExcel出力

KANA3サブフォーム役割

フォーム	推定される役割
KANA3A	データ入力・設定
KANA3B	計算パラメータ設定
KANA3C	波形表示・グラフ
KANA3D	統計結果表示
KANA3E	詳細分析
KANA3F	補正処理
KANA3G	MTT検査データのフロッピーコピー

2. KCDWプロジェクト

目的: 軌道変位データ処理・管理システム

構成ファイル

- メインフォーム: KCDW.frm (8,467行 - 最大規模)
- モジュール: myLib.bas, myLib2.bas, CmdLib.bas

主な機能

- DDB（Data Description Block）形式データ処理
- RSQ（Rail Sequence）データ管理
- 軌道変位の統計解析
- ピーク検出・異常値検出
- データベース連携（DAO使用）
- 複数のKCDW処理アルゴリズム実装

特徴

- 8,467行の大規模フォーム（最多）
- データベース処理が中心
- 高度な数値計算機能
- 6個のイベントハンドラで主要機能を実装

### 3. DCP2Sプロジェクト

目的: DCP形式データ変換ツール

#### 構成ファイル

- メインフォーム: DCP2S.frm, DCP2SA.frm
- モジュール: CmdLib.bas, mylib.bas, mylib2.bas

#### 主な機能

- DCP形式データの読み込み
- LABOCS形式への変換
- データ形式チェック
- バッチ変換処理

### 4. DCPZWプロジェクト

目的: DCP座標波形処理

#### 構成ファイル

- メインフォーム: DCPZW.frm
- モジュール: CmdLib.bas, mylib.bas, mylib2.bas

#### 主な機能

- 座標系データの変換
- 波形データの補正
- Z方向（高さ）、W方向（横）の処理

### 5. Ora2Lab2プロジェクト

目的: Oracle Database → LABOCS形式変換（通常版）

#### 構成ファイル

- メインフォーム: Ora2Lab2.frm (9,581行 - 最大)
- モジュール: mylib.bas, mylib2.bas, CmdLib.bas

#### 主な機能

- OracleデータベースからのデータSQL抽出
- LABOCS形式への詳細変換
- ADO/DAO接続
- Hgs専用ライブラリ使用
- データ検証・エラーチェック

#### 特徴

- ADO 2.5使用
- HgsLA/LZ系ライブラリ使用（独自通信プロトコル）
- 大規模データ処理対応

### 6. Ora2LaS2プロジェクト

目的: Oracle Database → LABOCS形式変換（簡易版）

#### 構成ファイル

- メインフォーム: Ora2LaS2.frm

- モジュール: CmdLib.bas, mylib.bas, mylib2.bas

#### 主な機能

- 簡易的なOracle→LABOCS変換
  - 基本データのみ対応
  - 高速変換
- 

## 主要フォーム説明

### KANA3.frm（簡易軌道波形パラメータ復元計算）

#### UIコントロール

- **Frame8:** MTT値入力エリア
  - Text7, Text8: 左レール BC/CD値
  - Text9, Text10: 右レール BC/CD値
  - Combo13, 15, 16: カント補正選択
- **Command1~14:** 各種処理実行ボタン
  - Command14: キロ程変換ツール起動

#### 主要イベント処理

- Form\_Load: 初期化处理
- Command1\_Click: 計算実行
- Command2\_Click: データ保存
- Command3\_Click: グラフ表示
- Command12\_Click: 設定
- Combo13\_Click: カント選択変更

#### 処理フロー

1. Form\_Load時にパラメータ初期化
2. ユーザーがMTT値、カント等を入力
3. Command実行で計算処理
4. 結果をテキストボックス/グラフに表示
5. Excel出力またはファイル保存

### KCDW.frm（軌道変位データ処理）

#### 主要処理

- DDB/RSQファイルの読み込み
- 軌道変位計算
- ピーク検出
- 統計値算出
- データベース更新

#### 特徴的な処理

- 102個の関数で複雑な軌道計算を実装
- データベーストランザクション管理
- バッチ処理対応

### DCP2S.frm（DCP変換）

## 処理概要

- Form\_Load: 初期設定
- ComboSenbetu\_Click: 線区選択
- Command1\_Click: 変換実行
- CommandEnd\_Click: 終了

## 変換フロー

1. 入力ファイル選択（CommonDialog使用）
2. 線区・区間指定
3. DCP → LABOCS形式変換
4. 出力ファイル生成

## Ora2Lab2.frm / Ora2LaS2.frm（Oracle変換）

### データベース接続

- ADO接続文字列設定
- SQL実行・レコード取得
- HgsLAClient/HgsLZClient使用（独自プロトコル）

### 変換処理

1. Oracle接続
  2. 軌道データクエリ実行
  3. LABOCS形式データ生成
  4. ファイル出力
- 

## 共通モジュール説明

### CmdLib.bas（コマンド処理ライブラリ）

規模: 33,843行、639以上の関数 役割: 全プロジェクトの中核となる処理ライブラリ

### 主要機能カテゴリ

#### 1. ファイル/I/O処理

主要サブルーチン：

- DDB軌道等ファイルからDDBを得る
- RSQ読み込み
- RSQ作成
- File文字列を文字列配列へ

#### 2. 軌道波形処理（BS系サブルーチン）

KCDW関連処理：

- Bs0510MKcdw: 基本軌道変位計算
- Bs0520MKcdw: 簡易波形処理
- Bs0530MKcdw: 値以上の内方処理
- Bs0540MKcdw ~ Bs0640MKcdw: 各種波形処理

DDB/RSQ処理：

- KCDW単軌DDB
- KCDW復軌RSQ復元



- KCDW復軌DDB復元

### 3. データ変換・計算

フィルタ処理：

- HPP1の波形のピーク読取
- KFLフィルタ前後計算
- KFLFFT（FFT処理）

統計処理：

- HMA一次移動平均処理
- K20波形20m弦の計算

### 4. 表形式データ処理

軌道形式データ：

- TD地点単位表ファイルから地点単位表配列へ
- TD区間系表ファイルから区間系表配列へ

KDT（キロ程データ）処理：

- KDT配列をKDTファイルへ
- KDTファイルからKdt区間配列へ

### 5. データベース関連

DDB（Data Description Block）操作：

- KUD軌道等DDBに標準値登録
- KUDNAMDDBにデータ名登録
- RSQ最大値平均値等をDDBに登録

### 6. その他ユーティリティ

ファイル存在チェック：

- FileExistMsg
- FileExists

文字列処理：

- GetWord
- DelSpc

### バッチコマンド方式

CmdLibは「バッチコマンド」として各処理を組み合わせて使用する設計。エラーハンドリングは IErr パラメータで統一。

## mylib2.bas（拡張汎用ライブラリ）

規模: 13,784行、238以上の関数

### 主要機能

#### 1. 型変換・文字列処理

- Clntw, CLngw, CSngw, CDbw: 文字列→数値変換（エラー処理付き）
- DelSpc2, DelSpc3, DelSpcAll: スペース削除
- DelCrtn: 改行削除
- DelZero: ゼロ削除

#### 2. ファイル操作

- FileExists, FileExists2: ファイル存在確認
- FileRecCount: ファイル行数取得
- FileDEL: ファイル削除
- FileKaraMojihaireru: ファイル→文字列配列
- MojihaireruKaraFile: 文字列配列→ファイル
- FileNameBunkatu: パス分解
- FIChgKaku: 拡張子変更

### 3. パス・ドライブ処理

- GetLbcPath: LABOCS パス取得
- GetHozonPath: 保存パス取得
- FdDrv: フロッピードライブ検出
- FdDriveReady: ドライブ準備確認

### 4. グラフィックス関連

- GetColorCode: 線色取得
- GetDrawStyle: 線種取得
- GetDrawWidth: 線幅取得
- GetFontBold: フォント太字判定

### 5. Excel連携

- ExcelExeName: Excel実行ファイル検索

## mylib.bas（基本汎用ライブラリ）

規模: 571行、34関数

### 主要機能

文字列処理：

- DelSpc: スペース削除
- Replace: 文字列置換
- StrCount, StrCut, StrIns: 文字列操作

トークン分割：

- TokenGet, TokenGet2: トークン取得
- TokenGet1: 最初のトークン

文字種判定：

- IsAllAlpha, IsAllDigit: 文字種判定
- IsAllNum, IsAlpha, IsDigit: 個別文字判定
- IsHankaku, IsZenkaku: 全角半角判定

ユーティリティ：

- Max, Min: 最大値・最小値
- Uswap: スワップ
- YesNoBox, YesNoCanBox: ダイアログ表示
- LenA: 文字列長（全角2バイト）

## KANA3lib.bas / KANA3lib2.bas

規模: 各874行、13関数

主な機能：

- StrToW: 文字列変換（KANA3専用）
- その他KANA3固有の処理関数

## KcdwKANA3.bas

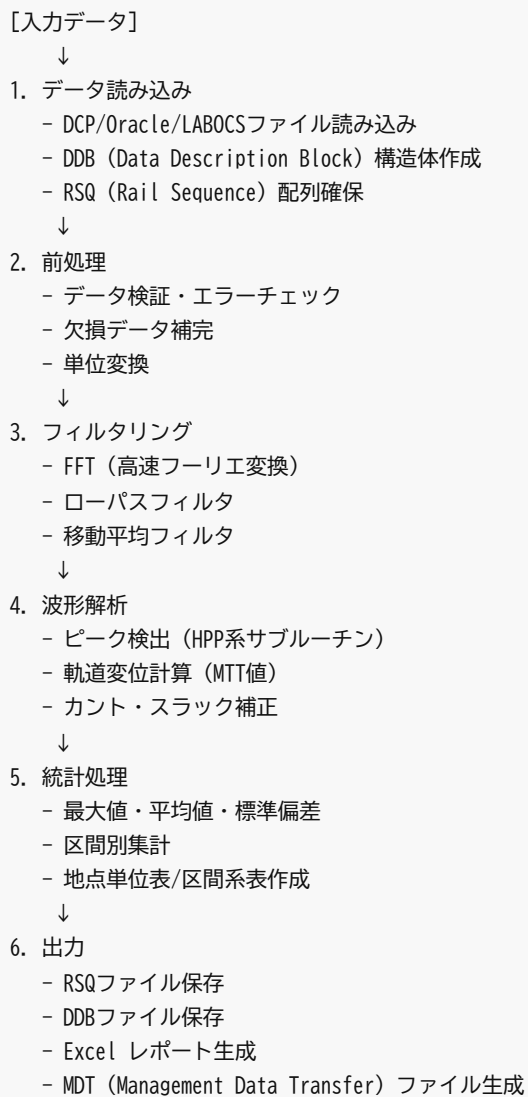
規模: 7行（定数定義のみ）

KCDW-KANA3間の連携用定数定義

---

## データフローと処理ロジック

### 典型的な波形復元処理フロー



### データ構造

#### DDB（Data Description Block）

軌道データのメタデータを格納する構造体。含まれる情報：

- 線路名、線路区分

- 測定日
- データ点数
- 開始距離、終了距離
- サンプリング間隔
- 最大値、平均値
- その他統計情報

#### RSQ (Rail Sequence) 配列

軌道波形データを格納する動的配列。測定点ごとの変位データを連続的に保持。

#### KDT (キロ程データ)

キロ程と測定データの対応表。含まれる情報：

- キロ程
- 測定値
- 補正值

---

## 波形復元アルゴリズム

### 1. FFT (高速フーリエ変換) 処理

実装箇所: CmdLib.bas 内の KFLFFT サブルーチン

処理内容:

1. 時系列波形データを周波数領域に変換
2. 指定周波数帯域でフィルタリング
3. 逆FFTで時系列データに戻す
4. ノイズ除去、平滑化に使用

### 2. ピーク検出アルゴリズム

主要サブルーチン:

- HPP1の波形のピーク読取: 単一波形のピーク検出
- HPP左右の波形のピーク読取: 左右レールのピーク検出
- HPP波形のピーク読取: 複数波形のピーク検出

アルゴリズム概要:

1. 移動窓 (ウィンドウ) を設定
2. 窓内の極値 (極大・極小) を検出
3. 閾値判定 (ABSYN="YES"で絶対値判定)
4. ピーク位置と値を配列に格納

### 3. 軌道変位計算 (Bs05系サブルーチン)

Bs0510MKcdw: 基本的な軌道変位計算

- KFIL: キロ程ファイル
- ZFIL: Z方向 (高さ) データ
- SFIL: スラックデータ
- KcdwFIL: 出力ファイル

Bs0520MKcdw: 複合軌道変位計算

- WFIL: W方向（横）データ追加
- XFIL: 補正データ追加

**Bs0530MKcdw: 高度な復元計算**

- DFIL: 微分データ
- HFIL: 高次データ
- KLR, TLR: 左右レール指定

**処理ステップ:**

1. 入力ファイル読み込み
2. キロ程マッチング
3. 左右レールデータ分離
4. カント補正適用
5. スラック補正適用
6. BC（Before Correction）/CD（Corrected Data）値計算
7. 統計値計算
8. 出力ファイル生成

## 4. HSJ系波形補正アルゴリズム

**HSJ5（区間復元の正面）:**

- HSJ5\_GETBC: 区間のBCデータ取得
- HSJ5\_SAIHI: 再帰的な補正処理

**処理内容:**

- 軌道の連続性を考慮した波形復元
- 異常値の自動除外
- 境界条件の適用

## 5. Y1Y2系処理

**主要サブルーチン:**

- Y1Y2F
- HsjY1Y2F
- Y1Y2

**機能:**

- 軌道の2次元的な波形解析
- Y1（縦方向）、Y2（横方向）の同時処理
- 曲線区間の補正

## 6. MTT値計算

**MTTパラメータ:**

- BC値：補正前データ（Before Correction）
- CD値：補正後データ（Corrected Data）
- カント補正係数
- スラック補正係数

**計算の流れ:**

MTT値 = (測定値 × 補正係数) + オフセット  
最終値 = MTT値 - カント補正 - スラック補正

MTT標準値（推定）：

- 左レール BC: 3.63
- 左レール CD: 9.37
- 右レール BC: 3.63
- 右レール CD: 9.37

## コード間依存関係

### プロジェクト階層構造

レベル1（基盤層）  
mylib.bas            - 基本文字列・ユーティリティ  
↓  
レベル2（拡張層）  
mylib2.bas           - 拡張ファイルI/O、型変換  
↓  
レベル3（ドメイン層）  
CmdLib.bas           - 軌道専用処理ライブラリ  
KANA3lib.bas          - KANA3専用処理  
↓  
レベル4（アプリケーション層）  
KANA3.frm            - KANA3メイン  
KCDW.frm             - KCDW メイン  
DCP2S.frm            - DCP変換  
Ora2Lab2.frm          - Oracle変換  
等々...

### モジュール参照関係

プロジェクト	mylib.bas	mylib2.bas	CmdLib.bas	KANA3lib.bas	KcdwKANA3.bas
KANA3	○	○	○	○	-
KCDW	○	○	○	-	○
DCP2S	○	○	○	-	-
DCPZW	○	○	○	-	-
Ora2Lab2	○	○	○	-	-
Ora2LaS2	○	○	○	-	-

### 関数呼び出し関係（主要な流れ）

[フォーム Command\_Click]  
↓  
mylib2.FileExists() でファイル存在確認

↓  
mylib2.FileKaraMojihaireru() でファイル読み込み  
↓  
CmdLib.RSQ読み込み() でRSQデータ取得  
↓  
CmdLib.HPP1の波形のピーク読取() でピーク検出  
↓  
CmdLib.Bs0520MKcdw() で軌道変位計算  
↓  
CmdLib.RSQ作成() で結果ファイル保存  
↓  
mylib2.ExcelExeName() でExcel起動  
↓  
[処理完了]

## 外部ライブラリ依存

全プロジェクト  
├─ stdole2.tlb (OLE Automation)  
└─ scrnun.dll (Microsoft Scripting Runtime - FileSystemObject等)

KCDW  
└─ DAO350.DLL (Microsoft DAO 3.51 - Database操作)

Ora2Lab2 / Ora2LaS2  
├─ msado25.tlb (ADO 2.5 - Database接続)  
├─ HgsLACCommon.tlb  
├─ HgsLZCommon.tlb  
├─ HgsLAClient.tlb  
└─ HgsLZClient.tlb (独自通信ライブラリ)

DCP2S / DCPZW / Ora2Lab2  
└─ comdlg32.ocx (Common Dialog Control - ファイル選択ダイアログ)

## 注意点と改善ポイント

### 1. VB6固有の課題

#### ランタイム依存

- VB6 ランタイム (msvbvm60.dll) が必要
- Windows 10/11では標準搭載されているが、将来的なサポート終了リスク

#### 推奨対応

- .NET Framework への移行検討 (VB.NET または C#)
- Python/JavaScript等モダン言語への書き換え

### 2. コードの保守性

#### 問題点

- 関数名が日本語 → 検索性・可読性が低い (エンコーディング問題も)
- 巨大なモジュール → CmdLib.bas 33,843行は分割すべき

- コメント不足 → 処理内容が不明な箇所が多い
- マジックナンバー → 定数化されていない数値が多数

#### 改善の方向性

- 関数名の英語化またはローマ字化
- モジュールの機能別分割
- コメント・ドキュメント追加
- 定数の定義ファイル作成

### 3. データ処理の効率性

#### 問題点

- 配列の動的確保 が頻繁 → メモリ断片化
- ファイルI/O が同期的 → 大量データ処理時に遅い
- トランザクション処理が不十分 → データベース更新時のエラーハンドリング

#### 改善案

- 配列サイズの事前計算と一括確保
- 非同期I/O・バッファリングの活用
- トランザクション処理の徹底

### 4. エラーハンドリング

#### 現状

エラーコード方式（古いスタイル）

- IErr パラメータで統一
- エラー原因が不明（IErr = 1 だけでは詳細不明）
- エラーメッセージが統一されていない
- ログ出力機能がない

#### 改善の方向性

- 構造化例外処理への移行（.NET移行時）
- エラーログの実装
- エラーメッセージの標準化

### 5. データベース接続

#### 問題点（Ora2Lab2系）

- ADO 2.5 は古い → ADO.NET 推奨
- 接続文字列がハードコード → 設定ファイル化すべき
- SQL インジェクション の可能性 → パラメータ化クエリ使用

#### 対応策

- パラメータ化クエリの徹底
- 接続文字列の外部化
- 最新のデータアクセス技術への移行

### 6. ハードコードされた値

#### 問題箇所

- MTT基準値（3.63, 9.37）
- ファイルパス
- データベース接続文字列



- 補正係数

#### 改善案

- INIファイルまたはXML/JSON設定ファイルに外部化
- 定数モジュールの作成

## 7. 文字コードとファイルパス

#### 問題点

- Shift-JIS エンコーディング依存
- Windows パス区切り（¥）固定
- 長いファイルパス（MAX\_PATH = 260文字制限）

#### 対応

- UTF-8 への移行
- パス処理の標準化
- UNCパス・長いパス対応

## 8. テスト・品質保証

#### 現状の課題

- 単体テストがない
- テストデータが不明確
- リグレッションテストの仕組みがない

#### 推奨事項

- ユニットテストフレームワーク導入
- テストデータセット作成
- CI/CD パイプライン構築

## 9. ドキュメント不足

#### 必要なドキュメント

- ☐ ユーザーマニュアル（操作手順）
- ☐ システム設計書（アーキテクチャ図）
- ☐ データベーススキーマ定義書
- ☐ API仕様書（関数リファレンス）
- ☐ 運用手順書（バックアップ、障害対応）
- ☒ ソースコード説明書（本ドキュメント）

## 10. セキュリティ

#### 潜在的リスク

- パスワードがハードコードされている可能性
- SQLインジェクションの脆弱性
- ファイルアクセス権限が不適切な可能性
- ログに機密情報が出力される可能性

#### 対策

- パスワードの暗号化保存
- パラメータ化クエリの徹底
- 最小権限の原則（ファイル・DB）

- 機密情報のマスキング

## 移行・モダナイゼーション推奨事項

### 短期対応（1-3ヶ月）

- コードレビュー - 重要な処理箇所の詳細解析
- テストケース作成 - 現行動作の記録
- ドキュメント整備 - 本説明書の詳細化
- バックアップ体制 - データとコードの定期バックアップ

### 中期対応（3-12ヶ月）

- VB.NET 移行 - .NET Framework 4.8 または .NET 6/7/8
- リファクタリング - モジュール分割、関数名英語化
- データベース最新化 - ADO.NET、Entity Framework
- 設定ファイル外部化 - App.config / appsettings.json

### 長期対応（1-2年）

- Webアプリ化 - ASP.NET Core, Blazor, または React/Vue
- クラウド対応 - Azure, AWS 等へのデプロイ
- マイクロサービス化 - 各プロジェクトを独立したサービスに
- API化 - RESTful API / GraphQL で他システム連携

## 付録

### A. 主要なBS系サブルーチン一覧

サブルーチン	処理内容
Bs0510MKcdw	基本軌道変位計算
Bs0512MKcdw2	軌道変位計算（拡張版2）
Bs0513MKcdw	軌道変位計算（拡張版3）
Bs0514MKcdwSA	軌道変位計算（SA版）
Bs0515MKcdw	軌道変位計算（拡張版5）
Bs0520MKcdw	簡易波形処理
Bs0530MKcdw	値以上の内方処理
Bs0540MKcdw	波形処理40
Bs0545MKcdw	波形処理45
Bs0550MKcdw2	波形処理50-2
Bs0560MaKcdw	波形処理60Ma
Bs0560MbKcdw	波形処理60Mb
Bs0570MKcdw2	波形処理70-2

Bs0585MKcdw	波形処理85
Bs0600MKcdw	波形処理600
Bs0610MKcdw	波形処理610
Bs0630MKcdw	波形処理630
Bs0640MKcdw	波形処理640

## B. 主要なHPP系（ピーク検出）サブルーチン

サブルーチン	処理内容
HPP1の波形のピーク読取	単一波形のピーク検出
HPP1の波形のピーク読取2	単一波形のピーク検出（拡張版）
HPP左右の波形のピーク読取	左右レールのピーク検出
HPP左右の波形のピーク読取2	左右レールのピーク検出（拡張版）
HPP波形のピーク読取	複数波形のピーク検出
HPP前後等のピーク抽出	前後関係を考慮したピーク抽出
HPP区間連続の波形のピーク読取	区間連続的なピーク検出
HPP区間連続の要注意箇所抽出	要注意箇所の抽出

## C. ファイル形式

### DCP形式

- 軌道検測車の生データ形式
- バイナリまたはテキスト
- Z（高さ）、W（横）、S（スラック）、K（キロ程）等のチャンネル

### LABOCS形式

- JR西日本の軌道管理システム標準形式
- テキストベース
- ヘッダー + データブロック構造

### RSQ形式

- Rail Sequence（レールシーケンス）
- 測定点の連続データ
- DDB（メタデータ）+ データ配列

### MDT形式

- Management Data Transfer
- 管理データ転送用
- テキストベース、複数レコード構造

### T3形式

- 新軌道T3形式
- バイナリ

- ヘッダー + 10k単位データ

---

## 改訂履歴

版	日付	改訂内容	作成者
1.0	2025-10-15	初版作成	AI Assistant

---

## 連絡先・参考情報

- 開発元: 技術部（推定）
- 使用部署: JR西日本 軌道管理部門（推定）
- 関連システム: LABOCS、KCDW、Oracle軌道データベース

---

本ドキュメント終わり