

障害者自立支援機器等研究開発プロジェクト

音声コード読み取り作成に関するソフトおよび携帯型機器の開発

平成21年度 総括研究報告書

研究代表者 金子 秀明

平成22（2010）年4月

第1章 音声コード読み取り作成に関するソフトおよび携帯型機器の開発	5
A.研究開発目標.....	5
B.研究開発方法.....	7
C.研究開発結果.....	21
D.考察.....	25
E.結論.....	25
F.健康危険情報.....	26
G.研究発表	26
H.知的財産権の出願・登録状況.....	26
第2章 エンコード（音声コード生成）プログラム	27
1 はじめに	27
2 ファイル構成・動作環境.....	27
2-1 ファイル構成.....	27
2-2 動作環境.....	28
3 機能説明.....	28
3-1 機能概要.....	28
3-2 SPコード作成（エンコード・シンボル化）	29
3-2-1 SpeechioEncode 関数	29
3-2-2 SpeechioSymbol 関数	30
3-3 音声コード読取機器の音声出力の設定（音質・音程・音量）	31
3-4 Microsoft Word VBA を利用した文書の設定.....	31
3-4-1 文単位に音声出力の設定.....	32
3-4-2 ページ毎に一つのシンボルを作成.....	32
3-4-4 シンボルのリサイズ.....	32
3-4-4 シンボルの貼付け位置の調整.....	33
3-5 シンボル・テキストの出力	33
3-5-1 シンボルの出力.....	33
3-5-2 テキストの出力.....	34
4 種類.....	34
4-1 ワンクリック版.....	35
4-2 バリアフリー版.....	36
5 Word マクロの開発.....	38
5-1 開発時のファイルの位置.....	38
5-2 エディタの起動.....	38

5-3	マクロの実行.....	39
5-4	マクロの配布.....	39
5-5	音声コード読取機器における「読み」の指定について.....	40
6	アプリケーション.....	40
6-1	ファイル構成.....	41
6-2	開発環境.....	41
6-3	動作環境.....	41
6-4	目的と機能.....	42
7	ユーザーインターフェイス（メニューとツールバー）.....	43
7-1	機能説明.....	44
7-1-1	機能.....	44
7-1-2	画面.....	44
7-2	詳細設定.....	44
7-2-1	機能.....	44
7-2-2	画面.....	45
7-3	読みの確認.....	46
7-3-1	機能.....	46
7-3-2	画面.....	46
7-3-3	音声合成ライブラリ.....	46
7-3-4	音声出力用のテキスト整形処理.....	47
7-4	環境設定（SP コード作成/読みの確認）.....	48
7-4-1	機能.....	48
7-4-2	設定内容.....	49
8	プログラム構成.....	52
8-1	Speechio.dot.....	52
8-2	SP_AUTH.dll.....	52
8-3	Speechio.dll.....	53
9	設定ファイルの構成.....	53
10	Word アドインの開発について.....	54
10-1-1	開発時のファイルの位置.....	54
10-1-2	エディタの起動.....	55
10-1-3	マクロの実行.....	55
10-1-4	マクロの設定.....	55
10-1-5	アドインの保護.....	56
第3章	デコード（データ<音声>出力部）プログラム.....	57
1-1	音声出力.....	57
1-1-1	処理の流れ.....	57

1-1-2 音声出力用データ作成	58
1-1-2-1 改行コード統一	58
1-1-2-2 文末判定	58
1-1-2-3 ポーズ変換	59
1-1-2-4 機種依存文字	59
1-1-2-5 読み仮名指定	59
1-1-2-6 SP 制御コード全文頭付加	60
1-1-2-7 データ末判定	60
1-1-3 データ出力	60
1-2 テキスト出力	61
1-2-1 処理の流れ	61
1-2-2 テキスト出力用データ作成	61
1-2-2-1 SP 制御コード削除	61
1-2-2-2 読み仮名指定	61
1-2-2-3 データ末判定	62
1-2-3 データ出力	62
1-3 関数一覧	62
1-3-1 音声出力	62
1-3-2 テキスト出力（画面表示）	63
2 デコードプログラム生成	63
2-1 動作環境・開発環境	63
2-2 I/F 仕様	64
2-2-1 画像データからの SP コード取り出し	64
2-2-2 SP コードデータからのテキスト抽出	65
3 SP コード抽出処理詳細	65
3-1 ReadSymbol0関数	65
3-2 全体の流れ	66
3-3 各部詳細	66
4 各処理の詳細	69
4-2 第一コーナーポイントの探索	71
4-3 シンボル外形の各ポイント探索	77
4-4 見出し線位置の探索	82
4-5 交差点の計算	84
4-6 ビットサンプリング	85
5. QR コードの取り込み	86
5-1 処理の流れ	86
5-2 QR コードの読み取り	87

第4章 音声コード読取機器の開発	88
1 PC 接続型	88
1-1 PC 接続型機器：今後の展開	91
2 携帯型	92
2-1 仕様	92
2-2 携帯型機器：今後の展開	93
付録1 SP Code 作成ソフトウェア開発者向け仕様書 (Speechio.dll)	95
1 SP コードの形状	95
1-1 各部の名称	95
1-2 サイズとデータ容量	96
2 ライブラリ (Speechio.dll)	97
2-1 入力仕様	97
2-2 エンコード関数：SpeechioEncode	97
2-3 シンボル作成関数：SpeechioSymbol	98
2-4 誤り訂正	99
2-5 関数の使用例	99
3 SP コード画像の印刷	100
3-1 画像の大きさ	100
3-2 画像の位置と余白	100
4 入力テキストの整形処理	102
4-1 制御コード等の削除	102
4-2 読みの指定	102
4-3 音声コード読取機器での声質の切り替え	103
5 その他条件	105
5-1 コードの位置を表す切り込みについて	105
5-2 用紙の指定について	105
5-3 プリンタについて	106
5-4 SP コードのコピーについて	106
付録2 音声出力用ユーザー辞書登録内容	107

第1章 音声コード読み取り作成に関するソフトおよび携帯型機器の開発

研究代表者 金子 秀明 (株式会社日本テレソフト 代表取締役)

研究要旨

視覚障害者、高齢者、失読症者の情報入手の支援を行なう機器の開発を行う。まず、視覚障害者の中でもスクリーンリーダー等の進歩により普及してきているパソコン対応のものを開発した。これは、読み取った音声コードやそれを文字化した情報を多く保存できる為、また、弱視の人にとってはパソコン(以下「PC」と記載)の画面に大きな文字を表示させて読む事ができる等、多彩な活用が出来る為である。次に、どこでも簡易に使用できるように小型機器(PDA 端末)に読み取り機能を付加した機器を開発する。このために専用スキャナ、携帯型機器およびそれに関連したソフトを開発した。

今回のプロジェクトでは、大きく分けて下記項目を主に行った。

- ①テキストデータを音声コードに変換し、印刷する機能(エンコード)
- ②印刷された音声コード、QRコード、一次元バーコードを読み取りテキストデータに変換し音声化する機能(デコード)
- ③その音声コードを読み取る機器の開発
 - ・PC接続型機器
 - ・携帯型機器

上記②では、読み込んだ音声コードの画像データに対し、画像の傾き補正、画像の輪郭調整、明暗のコントラスト補正を行うことにより読み取りの認識率を上げる事ができた。

既存にある専用の音声読み上げ機器(スピーチオ、テルミー)では、用紙を機器の用紙挿入口に差し込む方式を取っており視覚障害者にとっては真っ直ぐな方向に差し込むというのはとても操作しにくいもので、用紙が10度程傾いた角度で入った場合はSPコードも当然傾いてしまうので、認識率が良くなかったが、SPコードの傾きの補正強化により認識率に大きな効果が見られた。

インクジェットで印刷した場合インクの吹き付けによる輪郭の歪みや不鮮明な部分がありその輪郭を鮮明に調整するシャープ補正や、白色度68%の用紙に印刷した場合、明暗のコントラスト補正の強化により認識率を上げる事ができた。

また、音声コードのエンコード、デコードには、韓国で普及している新たな音声コード「新ハイスペックコード」にも対応させた。この音声コードは、18mm角のSPコードに納める文字情報量が800文字に対し、同じ18mm角の場合1,500文字の情報量を納める事ができる。

A.研究開発目標

視覚障害者や、文字の分からない者（失読症、外国人等）が、音声コードを付けた印刷物を音声コード読み取り機器を利用し、音声コードを読み取る事によって、文字情報を音声で得られる事を目標とする。

まず、PCで作成された文字データからSPコードの作成だけではなく、新ハイスペックコードの作成・印刷するエンコードプログラムの開発を行う。

次に音声コードを読み取ってテキストデータに変換し音声出力や、画面表示する（弱視向け拡大文字の表示機能）デコードプログラムの開発を行う。現状の音声コードの認識率は、高密度のレーザープリンターの印刷（600dpi以上）でないと認識できなかった。今回の開発は、インクジェットプリンター含めた印字密度（dpi）が300 dpi程度でも読める様読み取りレベルのアップを行う。

また、視覚障害者が扱うという観点から読み取った時の傾きがあっても（逆さに読み取っても）認識できる事を目標に行い、画像補正の強化、高性能のカメラを使う事で確実に読み取るデコードプログラム、機器の開発を行う。

開発する機器に関しては、これまでの専用機器（スピーチオ、テルミー）ではなく、小型PC上で処理しそのPCに音声コード読み取り用のスキャナを接続させるPC接続型を開発し、認識アップされた機能を組み込んだ機器（PC接続型）、更に、より障害者自立支援機器とする為に外出先でも利用できる、携帯型の機器を開発する事を目標とした。

PC接続型に関しては、弱視向け対応としてPC画面には反転させた文字（背景を黒、文字を白に表示）や、文字の大きさ調整の機能を組み込み見やすい文字の表示ができる事を目標とした。

携帯型機器に関しては、片手で楽に持て、軽いものにする為、幅10cm、奥行き5cm、高さ3cm程度で重さ200g以下のものを開発する事を目標とした。

また、現在、広まっている1次元バーコード、QRコードの音声化の機能も取り入れる事により、多様なコードに対応する事で多くの使う場が増し、音声コード利用への促進を図る機器となる事を目標とする。

1次元バーコードを読み取り音声化できるという事は、バーコードと関連付けられた商品名やメーカー名の情報が直ぐに音声情報で確認できるという事になるので、このバーコード情報取得の仕組みを考え、バーコードと商品情報の管理をしている組織と連携して行けば今後、大きな市場となりえるのでこの1次元バーコードの読み取り機能を組み込む事は大きい。

B.研究開発方法

障害者自立支援機器等研究開発プロジェクト表

月	8月	9月	10月	11月	12月	H22年 1月	2月	3月
1. アプリケーション、小型スキャナの基本設計	8月6日～9月30日							
2. PC版、携帯型アプリケーションPGMの設計、作成	8月6日～3月20日							
3. 音声コードエンコーダーPGM作成 (PC版、携帯型)	9月6日～3月20日							
4. 音声コードデコーダーPGM作成 (PC版、携帯型)	9月16日～12月25日							
5. 小型スキャナの設計・開発	8月6日～1月31日							
	●10/12～ 国際福祉機器展にて試作機の紹介							
6. 携帯型への対応の研究、開発	●製品化							
	12月20日～3月20日							
7. 評価テスト (視覚障害者団体等へ計3回実施)	●1/14 ●1/25 ●2/20 ●3/21							

※PGM (プログラムのこと)

1. アプリケーションソフトウェアと小型スキャナの基本設計

①アプリケーションソフトウェアの、処理項目、処理の流れを上げ基本設計を行う。→2へ

- ・起動処理
 - ・メニュー表示処理及
 - ・画面表示処理、画面読上げ処理
 - ・入出力処理 (画像データの取り込み、音声出力、画面表示)
 - ・音声コードエンコーダー、デコーダープログラムへの連携処理
- 上記項目を上げ、それぞれのデータの流れ、処理方法をまとめた。

②小型スキャナの筐体、基板の基本設計を行う。→5へ

- ・小型スキャナのサイズ、重さ目標を上げそのデザインを行う。
- ・中に組み込む基板を制御するのに必要な部品を検討。

(CPU、RAM、コンデンサ、抵抗、トランジスタ、トランス、コネクタ、その他 IC)

2. アプリケーションソフトウェアの設計

新たな新ハイスpekコードを含む音声コード作成、読み取りのアプリケーションソフトウェアを組み込んだWindows対応のアプリケーションソフトウェアを設計した。

音声コード (SPコード、新ハイスpekコード)、QRコード、1次元バーコードの画像読み取り後、デコード処理、テキストデータを音声出力させるアプリケーションソフトウェアの設計。

その他に基本設計で上げた各処理項目について順次設計を行い音声コード作成、読み取り動作の確認テストを行った。

また、これらアプリケーションソフトウェアは携帯型機器に対応させる為に、Windows対応からWindowsCEで動作出来る様にプログラムの移植の為の設計も行った。

3. 音声コードエンコードプログラムの作成

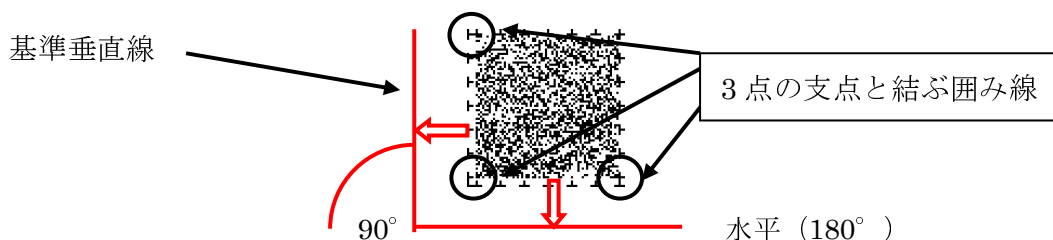
SPコードのエンコードプログラムの構築と新ハイスpekコードの日本語対応化のエンコードプログラムの構築を行った。QRコードに関しては、フリーのソフトが多数あったので、このプロジェクトでの開発から外した。

新ハイスpekコードは、情報量が多く、印刷条件などが緩和されており、実用上、大変に使いやすいが、日本語への対応ソフトウェアが未整備であったために、MS WordデータをこれまでのSPコードよりも文字数を多く格納できる音声コードフォーマット (新ハイスpekコード) にし、画像データに変換しMS Wordデータに組み込むソフトウェアを開発した。

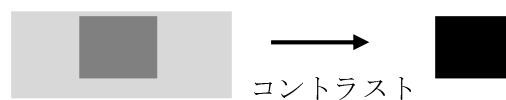
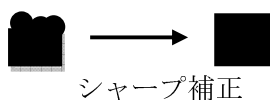
4. 音声コードデコードプログラムの作成

読み取った音声コードを画像データとして取り込み文字データに変換するデコードプログラムの構築を行った。

ここでは、画像補正処理の部分を改良し音声コード取り込み後、傾きや、色のコントラストの補正を行い、傾きに関しては、 180° (逆さま) になっても3点の支点とその3点を結んだ囲み線を認識し、その囲み線の左側と下側の線が基準垂直線に合う様に補正する事で認識可能となった。(逆さまの場合は、囲み線が右に来るので左にくるよう 180° 回転させてから基準垂直線に合わせる。)



また、インクジェットでの印刷の場合は、シャープ補正により、歪んだ輪郭を鮮明な線に変え、用紙の白色度が少ないものは、白黒の部分強調させ、取り込んでデコードさせる事ができた。



このプログラムの構築により従来の SP コード読み取り機器（スピーチオ）では、インクジェットで印刷したものや再生紙への印刷したものには、ほとんど対応できていなかったが、このシャープ補正、コントラスト処理により、読み取れる条件の制限が少なくなった。

インクジェットでは、メーカーの違いや、インクジェットのモデルにもよるが 300dpi, 用紙では新聞紙の材質に印刷されたコードの読み取りも出来るようになった。

その他に新ハイスペックコードのデコードプログラムの日本語化の開発、QR コードのデコードプログラムの開発を行った。

読み取り精度（○は認識できた。△は認識する時もあるが、しない時もある。×は認識出来ない）

印刷種類	紙質	スピーチオ	今回の開発した機器
レーザープリンター 600dpi	白色度 90%	○	○
	白色度 68%	○	○
インクジェットプリンター 600dpi	白色度 90%	△	○
	白色度 68%	×	○
印刷物 (オフセット墨単色)	白色度 90%	○	○
	白色度 68%	今回、実施せず	今回、実施せず

デコードプログラム内の音声対応については、日本で普及している音声コード（SP コード）読み取り機器に搭載している音声エンジンと同一の物を組み込むことにより、既存機器との「違和感」が出ないようにした。

新たに、違う音声エンジンを追加し、音声エンジンを切り替える事ができるので、各々が自分の聞き易い、好きな音質の音声を選択する事ができるようになった。

また、PC接続型の場合、デコード処理後のテキストデータの画面表示部分に関しては、拡大文字や、文字反転（背景色を黒に文字を白）、認識しやすいフォントへの変更の機能を盛り込み弱視障害者にも対応できる様に設計した。

5. 小型スキャナの設計・開発

音声コードを読み取り、文字情報に変換する為の機器として、小型のCMOSカメラを組み込んだ、QRコード、SPコード、新ハイスペックコードの読み取り可能な、小型スキャナの設計及び関連する制御プログラムの開発を行った。ここでは、カメラの当てる高さを変更し、光源調整を行う為に、回路図の設計やスキャナの設計を行った事によりSPコード、QRコード、新ハイスペックコードなどの2次元コードの読み取りを可能とした。

また、扱う視覚障害者、高齢者の方への配慮として機器操作が扱いやすい様、軽量小型化を図り、簡単に操作できる機器となる様設計を行った。

軽量小型化のスキャナ本体の目標を 50 g 以下とし、軽量化実現の為に、基板の実装 IC、電源部品のサイズを選定し小型基板を設計し小型スキャナを開発した。



PC 接続型(14 頁の「対象の機器」参照)

2次元コードを読み取る専用のスキャナとして、韓国製のスキャナがあり、それをベースに接続部分の改造や筐体のコードを読み取る接写面の改造を行った。

①USB接続について

既存製品は、シグナルコネクタとなっており、PC用との接続には変換アダプターを付ける必要があったので、直接USBに対応出来る様に接続部分（インタフェース部分）を改良した。

また、関連してWindows上でUSB接続によりスキャナ認識させる為のドライバーソフトウェアを開発してPC対応のスキャナとした。

これには、SPコードの特許を持つ、廣濟堂スピーチオ販売社などの協力を得て、基本ソフトの開示を受け、これを基にして専用のドライバーソフトウェアを開発した。

②SPコード対応のためのスキャナの改造



韓国製のスキャナでは、コントラスト不足やコード全体のサイズが大きく読み取りに収まらない為、SPコードを確実に読み取る事ができなかった。

そこで、コントラストの調整の為に光源のアップを行い、焦点調整の為にスキャナの接写部分に下駄を履かせカメラ位置の調整を行った

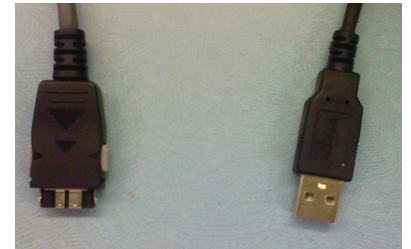
（上部写真：右は既存製品、左がSPコード対応カメラ）

スキャナの筐体部分の改造は下駄を履かせ高さの調節を行ったものを試作機とし、10月に開催された国際福祉機器展2009で紹介し、製品化した。

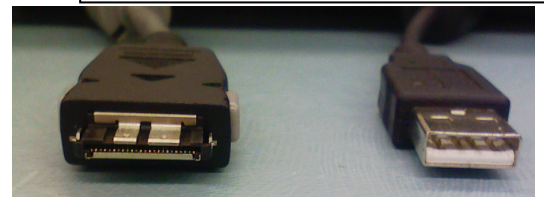
その後、他の2次元コード新ハイスペックコード、QRコードの読み取りに対応したソフトウェアの開発を行った。

6. 視覚障害者用のPC操作画面の開発

PC接続型機器はWindows対応となっているが、Windowsの組み込みシステム（エンベデッド：Embedded）で対応する事によりSPコード専用で使用するPCとして余計な常駐プログラムを削除しPCの起動短縮、処理時間の短縮を図った。



（右がUSB対応、左がシグナルコネクタ）

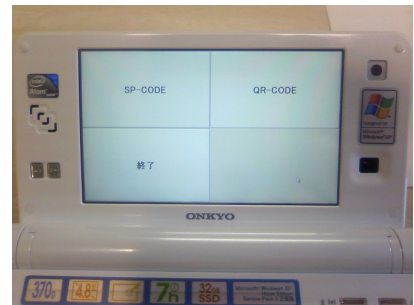


また視覚障害者が使いやすいようにキーの使用を制限して、誤動作を少なくするために画面操作を選択し易いレイアウトを考え、タッチパネル操作の選択メニューとした。

また、QRコード、SPコードを自動識別させる機能を検討したが、実際に自動認識するまでに10秒以上の時間がかかった為、自動識別のステップを組み込むのをやめ、QRコード、SPコードを選択して読みに行くようにした。それにより認識の時間が2秒程度となった。

PC起動後、読み取り専用のスタート画面となる。
右上などタッチすることで起動、タッチした部分を音声案内する。

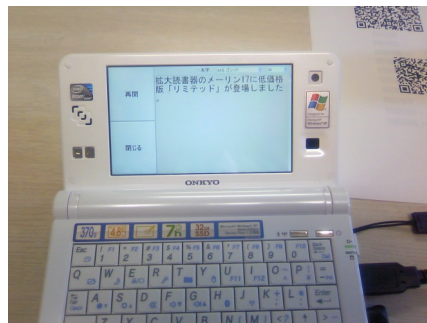
※起動時間を短縮し、他のカーソルキーを押しても、動作しないようにWindowsの不要コンポーネントを削除して使用することもできる様にした。



コード選択画面(タッチパネル式)



QRコードの読み取り画面



読み取りデコード後、文字と音声の出力となる。
左下をタッチすれば、終了となる。

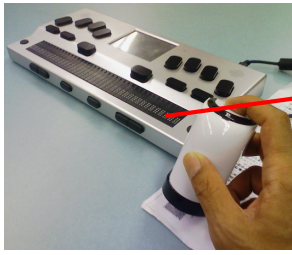
7. 携帯型機器への対応の研究

携帯型(小型)にする為に、システムの環境をWindowsCE, (Mobile)とした。その環境に対応する為のソフトウェアの開発、カメラ制御部分の構築を行ない、併せてWindowsCE環境を搭載する為の制御基板の設計、製作を行った。但し、その製作部分に関しては、委託先である、廣濟堂スピーチオ販売社が担当し、試作機の製作も担当した。この試作機にSPコードの読み取り音声化機能を組み込む事で、携帯型の対応を行った。

更に、より商品性の高い機能を考え、一次元バーコードを読み取り音声化をさせる機能を研究し実現した。

また、筐体部分に関しては、より商品化に近い型の試作機を作成する為に韓国にある既存の筐体を委託先の廣濟堂スピーチオ販売社が採用しSPコードに対応したプログラム等を組み込み試作機とした。

当初は、WindowsCEのPDAに、弊社既存製品である点字ディスプレイを組み込んだものがあり(次頁写真1)、これに廣濟堂スピーチオ販売社からのSPコードの仕様開示を受けWindowsCE向けの開発を進めていった。携帯型と言う意味では、一体化させた方が扱いやすく、片手で操作できる形状、重さを考え、次頁の写真2にあるようにカメラを組み込んだ機器を開発した。



点字表示部

写真1

SP コード及び点字ディスプレイ搭載型の PDA

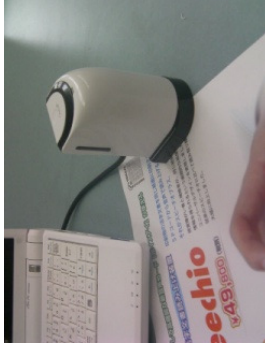
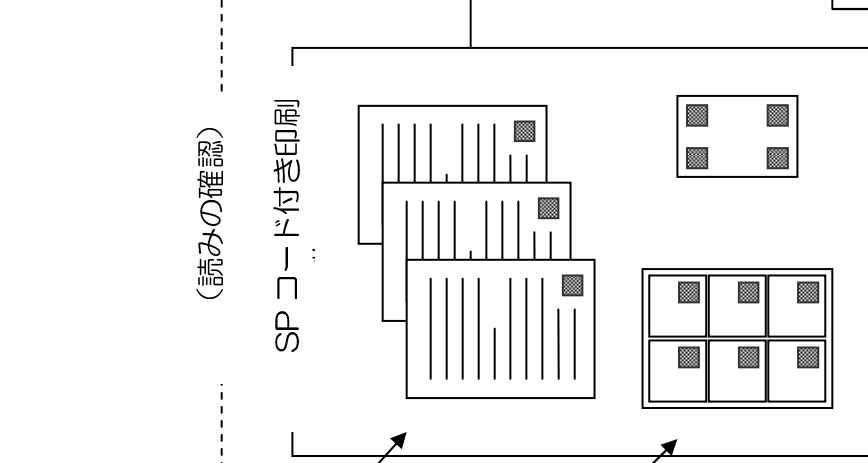
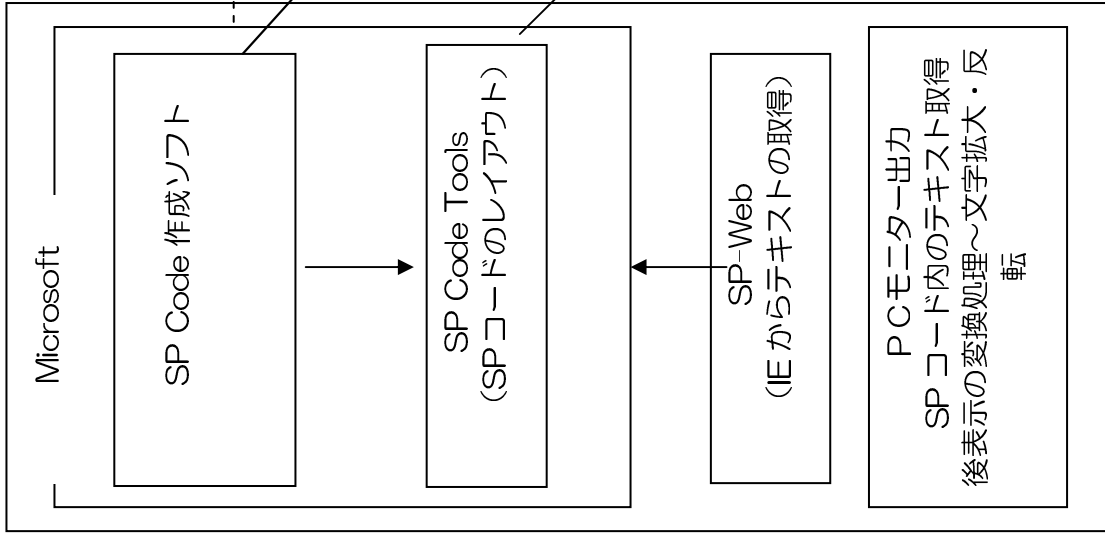


写真2

※詳細については14頁参照

音声コード作成・出力（音声・テキスト）の流れ

PC



※1 Microsoft Word 2000～2007 上で動作

8. 評価テスト

①概要

音声読み上げの機器について、スピーチオと開発した2機種の製品を使って評価してもらい、その操作性、識別度の確認などを、聞き取り調査した。

○対象の機器

- ・ 廣濟堂・スピーチオ



外形寸法 (幅) 100× (奥行き) 142× (高さ) 140mm

重さ 約 600g (電池含まず)

- ・ SpeechioPlus (今回開発した PC 接続型機器)



外形寸法 (幅) 50× (奥行き) 52× (高さ) 85mm

重さ 約 30g

(スキャナ部分のみ)

- ・ 携帯型・らいふ



外形寸法 (幅) 115× (奥行き) 63× (高さ) 27mm

※スキャナ部 (高さ) 40

重さ 170g

○モニター内容

SPコード記載の印刷物を提示し、2機種をそれぞれ使用してもらう。

使用した2機種は既存の機器スピーチオと開発した機器 **SpeechioPlus** である。

携帯型らいふは、3月21日のみモニター評価を実施。

2機種比較調査は、2010年1月、2月 計3回

3機種比較調査は2010年3月 1回

視覚障害者の方が中心なので、原則、聞き取り方式により、それぞれの評価と感想を聞く。

I 調査箇所

①佐賀弱視教育研究大会②熊本教科書点訳会（松崎悟代表）③日本点字図書館ミニ福祉機器展④名古屋盲人情報文化センター「見えない人・見えにくい人に役立つ第9回用具展」

II 調査項目

形状・重さ 操作性 読み取り精度 音質 商品性 自由回答

III 調査方法

視覚障害者団体での展示会にて、視覚障害者の方を対象に、レーザープリンタ、インクジェットプリンタ、オフセット墨単色印刷での印刷物の3種類の印刷物を対象に2機種或いは3機種を使用して操作していただき意見を聞く。

これらモニター評価を行う際の、倫理面での配慮については、「福祉用具臨床試験における研究倫理2008年度報告書」を参考にモニター対象者に対し、事前の説明と同意を得て行った。

まず、今回の調査の目的として、「国（厚生労働省）からの助成金で開発した、音声コード読み取り機器の機能について既存の機器と比較し、調査項目について3段階評価をし、今後の改良点、商品化へ向けての参考とさせていただく事」を説明した上で行った。

また、評価判断基準として既存の機器スピーチオと、開発した機器 **SpeechioPlus** との比較が主であったが、それ以外に実際に利用している音声関連の機器等とも比較した意見も聞いた。

モニターとなった対象者は展示会場に会場した視覚障害者の方とし調査項目を聞き取り方式をとり、調査データにはその回答、性別、年齢（年代回答）、視覚障害の状態（全盲、弱視）を反映させた。但し、個人情報に配慮し、それらの調査データについては、個人を特定できないように記録した。

IV 調査結果

1月14日（佐賀市）、1月25日（熊本市）、2月20日（東京都）、3月21日（名古屋市）、計4回

このモニターに協力いただいた方は58人で全てが視覚に障害をもっている方である。
その内訳は、全盲者 14人、弱視者 44人となった。

※障害区分の等級別での調査は行わなかった。

1月14日 モニター 26人（男性18人：女性8人）～第51回弱視教育研究全国大会
佐賀大会出展

	よ い	ふつう	わるい
形状・重さ	14	12	0
操作性	8	14	4
読み取り精度	20	6	0
音質	6	18	2
商品性	7	19	0

1月25日 モニター 5人（男性2人：女性3人）～熊本教科書点訳会

ここは、熊本教科書点訳会の代表である松崎悟氏に調査票*を基に調査を一任した。

***18頁参照**

	よ い (人数)	ふつう (人数)	わるい (人数)
形状・重さ	3	2	0
操作性	1	2	2
読み取り精度	4	1	0
音質	2	3	0
商品性	3	2	0

2月20日 モニター 10人（男性6人：女性4人）～日本点字図書館ミニ福祉機器展

	よ い (人数)	ふつう (人数)	わるい (人数)
形状・重さ	2	8	0
操作性	5	5	0
読み取り精度	8	2	0
音質	4	6	0
商品性	7	3	0

3月21日 モニター 17人（男性9人：女性8人）～名古屋盲人情報文化センター
「見えない人・見えにくい人に役立つ第9回用具展」

(人数)

	よ い (人数)	ふつう (人数)	わるい (人数)
形状・重さ	10	7	0
操作性	8	8	1
読み取り精度	4	8	0
音質	4	12	1
商品性	6	11	0

携帯型の調査（同日、同会場にて実施）

	よ い (人数)	ふつう (人数)	わるい (人数)
形状・重さ	9	6	0
操作性	2	10	5
読み取り精度	0	8	9
音質	6	13	0
商品性	14	3	0

総合評価

	よ い (人数)	ふつう (人数)	わるい (人数)
形状・重さ	29	29	0
操作性	22	29	7
読み取り精度	41	17	0
音質	16	39	3
商品性	23	35	0

*熊本教科書点訳会の調査でを使用した調査票

実施日 平成 年 月 日

音声コード読み取り機器性能調査票

調査担当 熊本教科書点訳会代表 松崎悟

音声コード読み取り機器従来型スピーチオと開発中の PC 接続型機器 SpeechioPlus の性能比較をし、下記の項目について従来型と比べた評価を回答下さい。

- | | | | |
|-------------|----|-----|-----|
| 1. 形状・重さ | よい | ふつう | わるい |
| 2. 操作性 | よい | ふつう | わるい |
| 3. 読み取り精度 | よい | ふつう | わるい |
| 4. 音質 | よい | ふつう | わるい |
| 5. 商品性 | よい | ふつう | わるい |
| 6. その他要望、意見 | | | |

参考

性別	年齢層	障害状態
男 ・ 女	代	全盲 ・ 弱視

モニター評価意見

①形状・重さ

- ・特にわるいと言った意見がなかった。小さく手で持っても丁度いい。
- ※既存機器は外出時等には、かさばって持ち歩きにくいですが、PCを含め、カメラもバック等におさまりやすいので持ち歩ける)
- ・軽い。 ※既存機器 600g に対し開発した機器 30g
- ・PC 接続型と言う面で、携帯性に欠ける。

②操作性

- ・読み取る時のボタンの感触がハッキリしないので、分かりづらい。
- ・ボタンが押しづらい。
- ・用紙角にカメラが正確にセットしなければならないという慎重さのストレスなく簡単に読み上げてくれる。
- ・PC が必要なので直ぐに読みたい時に OS の起動の時間がかかるので、直ぐに使えない。

③読み取り精度

- ・既存の機器に比べ、非常に精度が上がったと言う意見が多かった。
感覚的意見であるが、これまでは、5 割以下の認識率だったが、8 割以上には感じるとの事。
- ・インクジェットプリンターで印刷されたものが読めるのは非常に良い (印刷物を気にせず使える)
今後は印刷条件を気にせずに読ませる事ができるので、音声コードでの提供も楽になる。
これから普及が進むのではないか。
- ・音声コードの位置確認方法という根本的な問題の解決をしなければならないが、100%に近い精度は必要

④音質

- ・スクリーンリーダー等の視覚障害者用 PC 音声ソフトの音質と比べても大きな差が無いとの意見が多数。全く聞き取れないといった意見はなかった。
- ・既存の機器と比べても聞きづらさはないが、現在の PC 上の視覚障害者向け画面読上げソフトであるスクリーンリーダーの音質は、更によくなっているので同じくらいになると尚、良い。

この部分の変更に関しては、再度音声エンジンの検討、検討の際には、音声エンジンを始めから構築するか、聞き易いとされている音声エンジンを組み込むか、それぞれの費用、開発時間を比較し検討して改良するかを決める。

⑤商品性

- ・「わるい」との意見は無かった。
- ・操作性、形状（PC 接続型）にも関係してくるが、携帯性を持たせた方が良い。

総合評価の中で「わるい」が多かった操作性が全体の 12%を占めたので目標を 10%以下にするようスイッチを感触あるものに変更し（押した時に音がする）スイッチの感触を明確にさせた。その後の評価をしていないので実際の評価は得てない。

「形状・重さ」、「操作性」、「商品性」に関わった携帯型の要望に関し、当時は開発中で、2月までのモニター調査では、実際の携帯型機器の試作を見せる事が出来なかったため、それ以降に対応している。

3月の調査で実際にその携帯型機器を見てもらう事ができたが、SP コードのデコード部分の性能が PC 接続型に比べ CPU の処理能力が不十分であり、補正処理が十分に出来ず読み取り率が良くない状態での調査となった。読み取り部分については開発途中での評価となり今回は、形状・重さ部分の評価が中心となった。携帯性、重さ等の評価について既存の単体機器（廣済堂：スピーチオ、日本福祉サービス：テルミー）と比べると携帯性に関しては十分良い評価をいただいた。

また、携帯性以外にバーコードを読み取り音声出力する機能を見てもらい、実現を強く希望する声全員から上がった。

実際に評価できる段階になった時点で改めてモニター評価していきたい。

今後は、視覚障害者の他に、文字の見えづらくなった高齢者も対象とする事ができ、商品性、マーケットの拡大となる機器でもあるので、高齢者も対象とした調査を今後、行っていきたいと考えている。

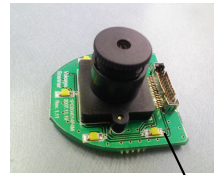
C.研究開発結果

1. 小型スキャナーの開発

SPコードを正確に読ませるため、スキャナーの改良（光源のアップや接写面の高さ調整）したCMOSカメラ搭載の小型スキャナーを開発（写真参照）し、USBでPCに接続させ取り込んだSPコードの画像データをPC上で解析し音声出力、文字出力が出来る様になった

また、今回、同じ様にQRコードも音声出力、文字出力が出来た。

文字出力は弱視障害者の方が見易い大きな文字、反転文字が出来る様になった。



CMOSカメラ搭載のスキャナ



全体図

スキャナーの外寸

幅100mm×奥行き142mm×高さ140mm

重さ 30g

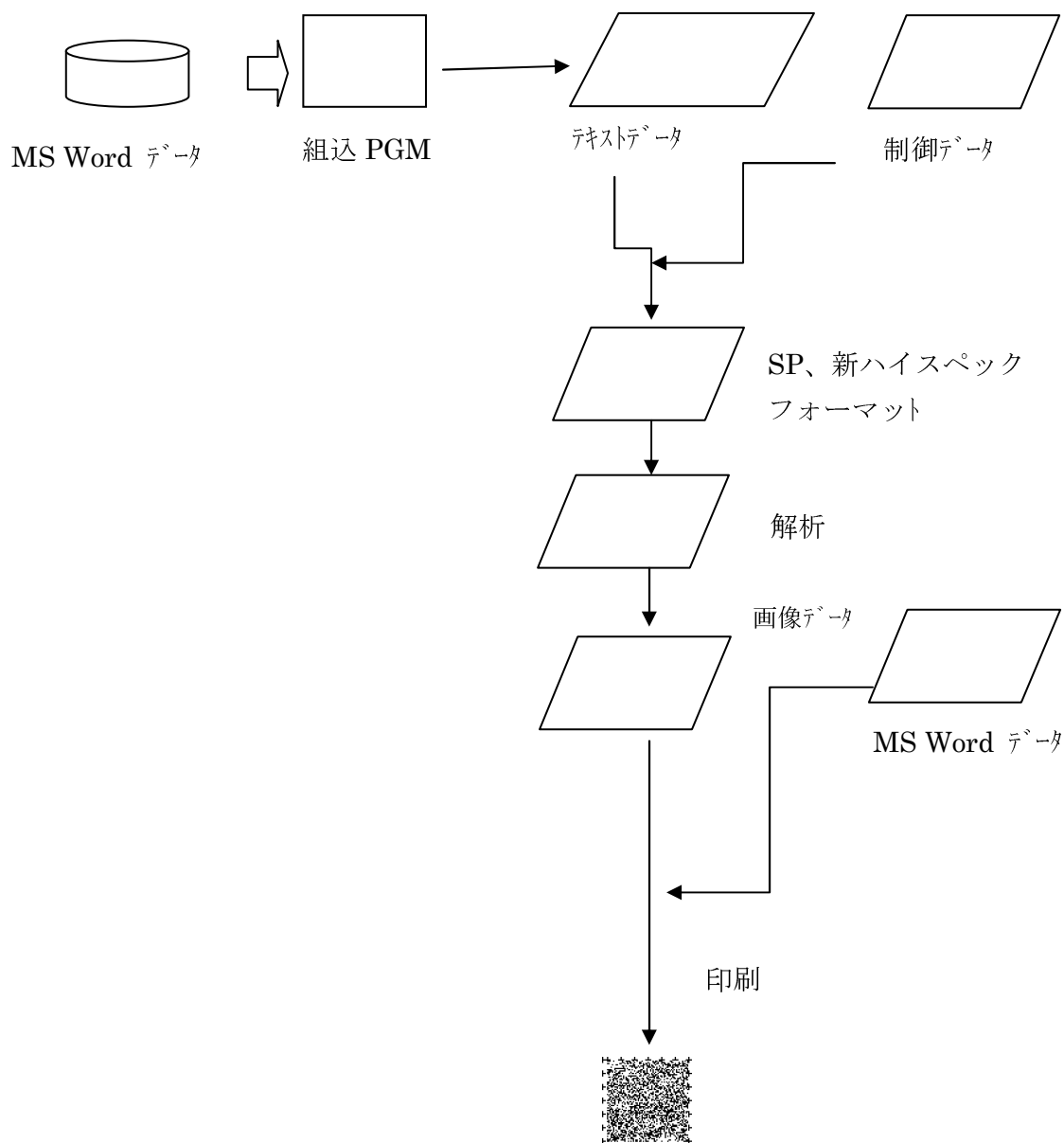
USBコードの長さ 1.5m

2. エンコード、デコードプログラムの開発

■エンコード処理

情報を多く格納できる新ハイスペックコードは韓国のハングル語と、英語にしか対応していなかったが、日本語に対応したエンコードプログラムの構築を行い、SPコードを含めたエンコードプログラムを開発した事により、2次元の音声コードを図形データとして組み込み印刷する事が可能となった。

MS WordデータをこれまでのSPコードよりも文字数を多く格納できる音声コードフォーマットにし、画像データに変換してMS Wordデータに新ハイスペックコードを、組み込めるようになった。



●フロー説明

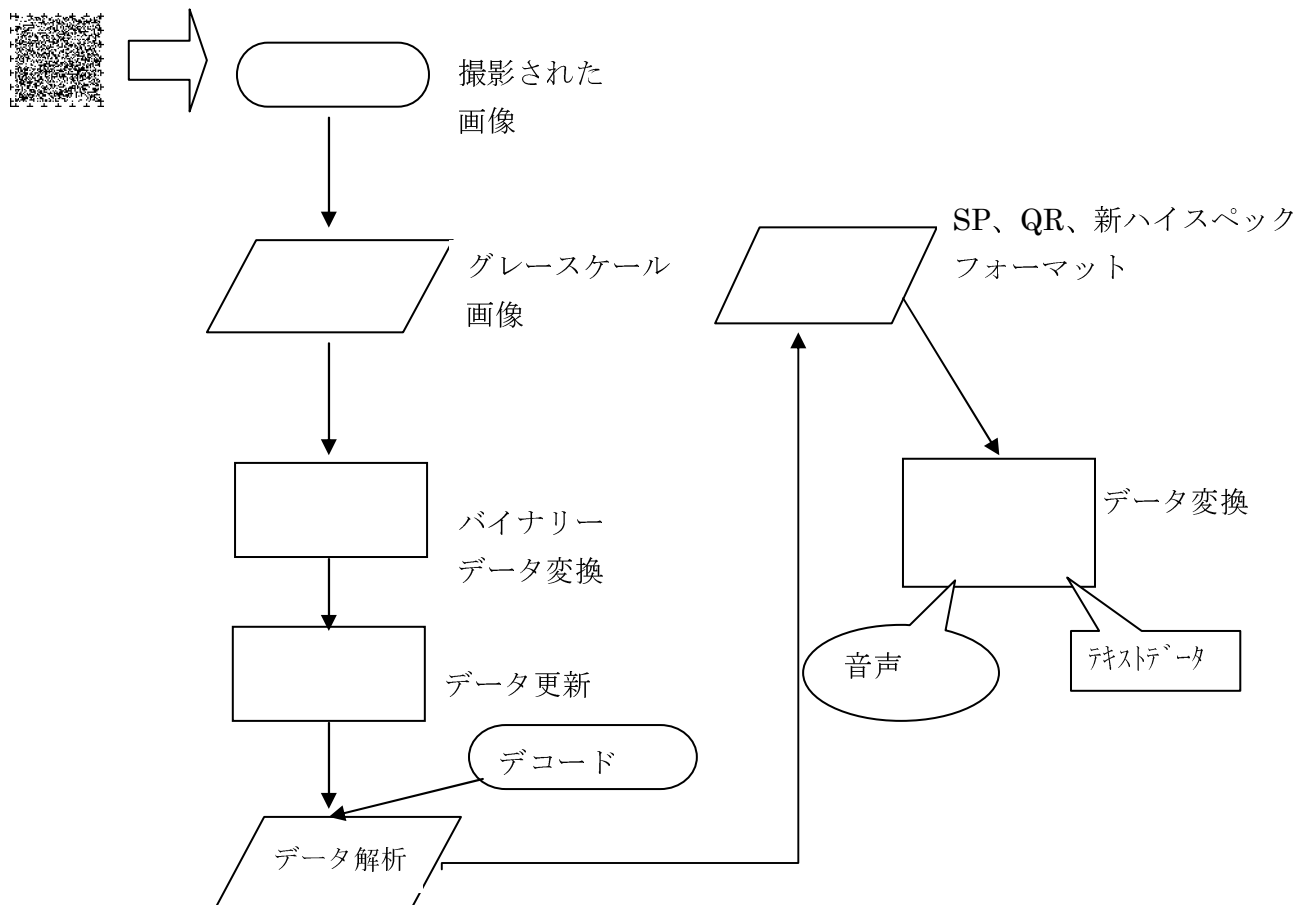
- ①Microsoft Word 文書を読み込む
- ②Microsoft Word のアプリケーションに組み込んだプログラムによりテキストデータに変換する。
- ③SP コード、新ハイスペックコード作成用の制御データをテキストデータに組み込む
- ④SP、新ハイスペックコードフォーマット生成
- ⑤SP、新ハイスペックコードフォーマットを解析し画像データに変換しMicrosoft Word 文書に所定の位置に貼付ける

■デコード処理

読み取った音声コードを画像データとして取り込み文字データに変換するデコードプログラムの構築を行い既存の音声コード読み取り機器（スピーチオ、テルミー）よりも読み取る条件の範囲がひろがった。

SPコードを読み取った後に、取り込んだ傾きや、色のコントラストの補正を行った事で傾きが、180°（逆さま）になっても認識可能となった。また、300dpi程度で印刷されたインクジェットでの印刷や用紙の白色度の少ない新聞紙などに印刷されたものを取り込んでデコードする事が可能となった。

その他に新ハイスペックコード、QR コードも読み取ってデコードする事が可能となった。



※SP コード、QR コードおよび新ハイスペックコードに対応。

●フロー説明

- ①音声コード、QRコード読み取り
- ②グレースケール画像に変換（併せて傾きやコントラストの補正を実行）
- ③画像データから解析したものをコンピュータ処理する為のバイナリーデータに変換しそのデータの配列を更新させる。
- ④更新したデータにデコードプログラムを介しSPフォーマット、QRフォーマット、新ハイスペックフォーマットとして解析・変換する。
- ⑤SPコード、新ハイスペックコード、QRコードのフォーマットを音声出力、テキストデータ出力させる。

3. 携帯型機器への対応

携帯型機器の開発の為、システムの環境をWindowsCE, (Mobile) とし、その環境に対応する為のソフトウェアの開発、カメラ制御部分の開発、WindowsCE環境を搭載する為の制御基板の設計を行い、携帯型機器の試作機を製作した。この試作機にSPコードの読み取り、音声化機能を組み込む事で、携帯型機器でSPコードが読める様になった。

更に、より商品性の高い機能を組み込む為、一次元バーコードを読み取り、音声化をさせる機能を追加し実現した。

結果、WindowsCE環境でSPコード、新ハイスペックコード、1次元バーコード認識から音声出力まで出来る様になった。

更に視覚障害者用付加機能を考えていく上で例えば、音声デジタルデータ*（デージーフォーマット）への対応等を考える事により、高機能な視覚障害者の日常生活にもなる。

また、携帯電話への対応だが、各社携帯電話に搭載されている基本ソフトウェアの対応に時間がかかるが、そのシステムに対応させるアプリケーションソフトウェアの最適な組み込み方法を検討し、今後の開発へ進めていく。



試作写真

*音声デジタルデータ（デージーフォーマット）とは、DAISYコンソーシアムが策定するアクセシブルな電子書籍フォーマット。視覚障害者や失読症でも読書が可能のように、文章を音声で読み上げたり、読まれている文章のハイライト表示が可能。米国や欧米の多くの国で障害者向け電子教科書や電子書籍のフォーマットとして採用されている。米国の標準規格 ANSI/NISO Z39.86-2005 として無償で公開されている。（日本国内も点字図書館を中心にネットワーク化が進み普及している）

D.考察

今回の開発プロジェクトにおいて「音声コード読み取り作成に関するソフトおよび携帯型機器の開発」を行った結果、既存の廣濟堂・スピーチオと比べ、印刷形態によって読めない、傾いた位置から読めないものが読めるようになった。

これは、SPコードの傾きやコントラストの補正処理が大きな効果となった。

また、一般に出ている1次元バーコード、2次元コード（QRコード）にも対応した事により、これまで以上に用途が広がると考えられる。

視覚障害者の方にとって日常必需品に近い機器となる。

今後の課題として音声コードの認識率を更にアップさせる必要があり、画像処理方法や、カメラ部分の性能等を研究して認識率アップ化を図りたい。

E.結論

この開発プロジェクト「音声コード読み取り作成に関するソフトおよび携帯型機器の開発」は、画像補正処理の向上により、従来の音声コード読み取り機器（廣濟堂・スピーチオ等）では、10°程の傾きがあった場合認識できなかったが、今回の開発により、180°（逆さま）に向きが変わっても認識できるようになった。また、読み取る印刷物もインクジェットで印刷したものでも認識出来るようになった。

傾き----- 逆転（180°）しても認識可能

印刷環境--- インクジェット（600dpi以上）、レーザー、一般的なオフセット
印刷したもの

携帯型の開発に関しては小型機器組み込み用OSであるWindowsCE（Mobile）で対応できた。また、今後はPC接続型にあるQRコードの対応や、音声デジタルデータ（ディジーフォーマット）再生機能、スキャナを当てると色を音声で案内する色認識等の付加機能を持たせ、音声コード、バーコードの認識精度面の向上を図り開発を進めて行く。この機器が商品化レベルまでいけば、視覚障害者にとって日常生活の機器として広がると考えられ、文字の見えづらい高齢者や失読症（ディスレクシア、外国人）等も対象になり市場も広がると考えられる。

また、1次元のバーコードを読み取りその商品名、メーカー名の情報の音声化が可能となったが、財団法人流通システム開発センターが管理している商品名、メーカー名の入ったJAN商品基本情報データベースの供給が必要となってくる。そのデータベースを取り込むみ、バーコードとそれに対応したデータベースのリンクが必要になる。今後は、財団法人流通システム開発センターの協力を得る必要があり、実現する事により普及は加速される。

F.健康危険情報

機器を使用して人体に影響を及ぼす危険性として上げるとすれば、電気部分になるが、電力はPC接続型の場合USBからのBusPower供給になるので2.5W以下となる。つまり、感電等により人体に影響を与える事は無く、携帯型に関してはバッテリー使用で一般に出回っている、デジタルカメラ、携帯電話等のバッテリーと同等なので危険性はない。

形状も、プラスチック素材で湾曲型に設計しており、鋭利な角度部分がないので、人体を傷つける様な部分はない。

G.研究発表

このプロジェクトでの研究発表は、行ってない。

H.知的財産権の出願・登録状況

未対応。

第2章 エンコード（音声コード生成）プログラム

1 はじめに

この音声コード読取機器で読み取り可能な SP コードを、Microsoft Word 上で作成するためのマクロであり、以下の機能を提供します。

- SP コード作成（エンコード・シンボル化）
- 音声コード読取機器の音声出力の設定（音質・音程・音量）
- Microsoft Word VBA を利用した文書の設定
- シンボル・テキストの出力機能

本章では、ソフトウェアの構成と各機能、処理の流れについて説明しています。

2 ファイル構成・動作環境

2-1 ファイル構成

本ソフトウェアは、マクロを記述した Microsoft Word 用のテンプレートファイル、エンコード・シンボル作成用 DLL ファイル、文字列処理用 DLL ファイルの3ファイルにより構成されています(ファイルのコピー先は Word および Windows のバージョンにより異なります)。



図1 ファイル構成図

※ 「SP_AUTH.dll」と「Speechio.dll」はシステムフォルダなど、パスの通ったディレクトリにあれば動作します。「Speechio.dot」は、Wordのバージョンやユーザ設定によりコピー先が異なります（「5. Word マクロの開発」参照）。

2-2 動作環境

Microsoft Word 2000～2007 上での動作を確認しています。Windows のバージョンは各 Word の動作環境に依存します。

3 機能説明

ここでは、エンコードプログラムの機能について詳細に説明する。

3-1 機能概要

本ソフトウェアは、音声コード読取機器で読み取り可能な SP コードの作成を目的とした Microsoft Word 上で動作するマクロであり、Word 文書のテキストデータを入力し、SP コード画像データを出力することを基本機能とします。以下に機能の概要図を示します。

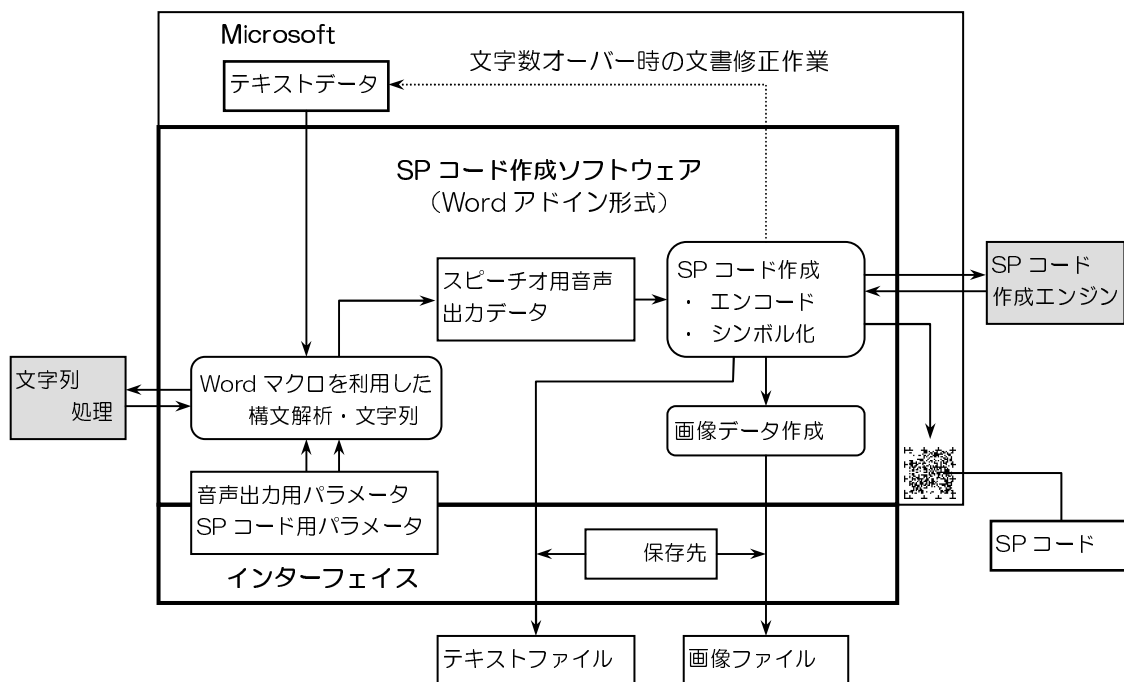


図 2 音声コード作成ツール機能概要図

また、音声コード読取機器の音声出力に関するパラメータの設定機能や、作成した SP コードを音声コード読取機器で読み込み可能な大きさと位置に変更する機能などを有します。

3-2 SP コード作成 (エンコード・シンボル化)

SP コードの作成には、「Speechio.dll」内にある `SpeechioEncode` 関数と `SpeechioSymbol` 関数を呼び出して、テキストデータのエンコードとエンコードデータのシンボル化を行います。

`SpeechioEncode` 関数では変換対象となるテキストデータがエンコードされ、エンコードされたデータは `bit_string` へ格納されます。ここで、`bit_string` は 0 と 1 の配列であり、0 が SP コードの白のセルを、1 が黒のセルをあらわします。この `bit_string` と、SP コードの縦横のセル数をもとに、`SpeechioSymbol` 関数によって、SP コードのイメージデータ (DIB 形式) が作成されます。作成されたイメージデータ (以下シンボル) はクリップボードへ転送されます。

3-2-1 SpeechioEncode 関数

<pre>short __stdcall SpeechioEncode(char data_type_string[], char cell_type_string[], char recover_level_string[], char dummy[], short data_size, unsigned char data_code[], char path_name[], unsigned char bit_string[])</pre>	
<code>data_type_string[]</code>	変換対象データの種別を指定します =T テキストデータ (日本語) =E テキストデータ (英語) =C コントロールカード作成用
<code>cell_type_string[]</code>	作成するシンボルサイズを指定します =s XS =S S =m M =M L =L XL
<code>recover_level_string[]</code>	誤り訂正の強さを指定します =S 大 (Strong) =N 中 (Normal) =P 小 (Poor)
<code>dummy []</code>	未使用

data_size	変換対象データのバイト数
data_code[]	変換対象データ
path_name[]	変換対象の文書があるディレクトリへのパスを指定します
bit_string[]	
戻り値	>0 成功 : bit_string へエンコードデータが格納されます =0 失敗 : エラー <0 データ容量オーバー : -1× (オーバーしたバイト数)

3-2-2 SpeechioSymbol 関数

<pre>short __stdcall SpeechioSymbol(short col, short row, unsigned char symbol_data[])</pre>	
col	シンボルの横のセル数
row	シンボルの縦のセル数
symbol_data[]	SpeechioEncode 関数で作成した、bit_string を指定します
戻り値	=0 成功 : 作成されたビットマップ画像はクリップボードへ転送されます =-1 失敗 : デバイスコンテキストの取得に失敗 =-2 失敗 : ビットマップ画像の作成に失敗

※1 1セルは4ドット×4ドットにより構成されます

※2 セル数はエンコード時のシンボルサイズに依存します (次表参照)

表 1 シンボルサイズとセル数

シンボルサイズ	col	row
XS	40	40
S	73	73
M	106	106
L	117	117
XL	139	139

3-3 音声コード読取機器の音声出力の設定（音質・音程・音量）

テキストデータの先頭に制御コード（SP 制御コード）を付加することで、音声コード読取機器の音声出力に関する設定を行うことが可能です。設定の種類と対応する制御コードは以下の通りです。

表 2 SP 制御コード一覧

音質	制御コード	種類・レベル
声質	^V0	男性音（デフォルト）
	^V1	女性音
高さ	^H0	レベル 0（最低）
	^H1	レベル 1
	^H2	レベル 2
	^H3	レベル 3（男性のデフォルト）
	^H4	レベル 4（女性のデフォルト）
	^H5	レベル 5
	^H6	レベル 6
	^H7	レベル 7（最高）
大きさ	^P0	レベル 0（最小）
	^P1	レベル 1
	^P2	レベル 2
	^P3	レベル 3
	^P4	レベル 4（デフォルト）
	^P5	レベル 5
	^P6	レベル 6
	^P7	レベル 7（最大）

※1 コントロールカード作成の場合、制御コードの付加は行いません

※2 制御コードは^V=0x16、^H=0x08、^P=0x10 をあらわします

3-4 Microsoft Word VBA を利用した文書の設定

処理の簡便化を図るため Microsoft Word VBA を利用して、文単位に音声出力の設定、ページ単位にシンボルの作成を行います。また、作成した画像を音声コード読取機器で読み取り可能とするため、リサイズ・位置の調整を行います。

3-4-1 文単位に音声出力の設定

Word VBA の Sentences オブジェクトを利用して、文書を文単位に分割します。各文のテキストデータの先頭に制御コードを付加することで、文単位に音声出力の設定を行うことが可能です。

※ 表組みの場合は、表の各セル内が 1 文として認識されます

3-4-2 ページ毎に一つのシンボルを作成

ページの区切りを位置情報として取得し、各ページ内の文章を一つのシンボルに変換します。

※ ページをまたぐ文については、以下の設定が可能です

- 前のページに含める
- 次のページに含める
- 表示どおりに分割
- 自動(ページをまたぐ文の中央位置が、ページの先頭よりも前であれば前のページに、後ろであれば次のページに含める)

3-4-4 シンボルのリサイズ

クリップボードへ転送された画像データを Word VBA の Shapes オブジェクトとして取り込み、次式によってシンボルサイズ (mm) を算出して、大きさを変更します。

$$\text{大きさ (mm)} = (\text{セル数}) \times 4 \times 25.4 / (\text{プリンタの解像度})$$

本ソフトウェアでは、プリンタの解像度を 600dpi としてサイズを算出しています。次の表は、その大きさ (mm) をあらわしたものです。

表 3 シンボルの大きさ (mm)

シンボルサイズ	大きさ (mm)
XS	6.8 × 6.8
S	12.4 × 12.4
M	17.9 × 17.9
L	19.8 × 19.8
XL	23.5 × 23.5

3-4-4 シンボルの貼付け位置の調整

リサイズして大きさの決定したシンボル画像を、音声コード読取機器の構造に合わせて位置の調整を行います。シンボルサイズによらずに、シンボルの中心位置が印刷物の右端から 25mm、下端から 25mm となるように移動します。

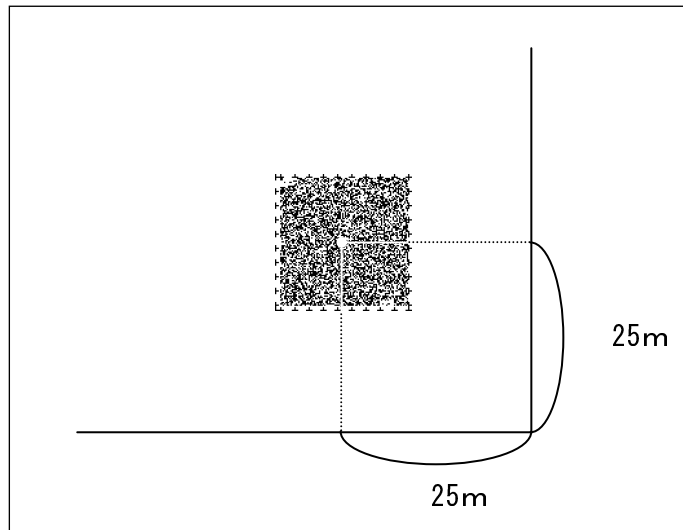


図 3 シンボル画像の貼付け位置

3-5 シンボル・テキストの出力

本ソフトウェアでは、Word 形式の文書内にクリップボードからシンボル画像を取り込んでいます。しかし、一般の印刷物では Word 形式が使用されることはないため、一般の印刷物への対応を考慮して、シンボルを BITMAP ファイルとして出力します。また、同時にエンコードする前のテキストデータをテキストファイルとして出力します。

3-5-1 シンボルの出力

シンボルの作成後、クリップボード内の DIB 形式のデータを BITMAP ファイルとして出力します。出力するかどうかはユーザが選択できるものとし、出力先もユーザが指定したフォルダとします。

ファイル名は、Word 文書のファイル名とページ番号から作成します。また、同名のファイルが出力先に存在する場合は、上書き保存します。

文書 1 . doc → 文書 1 _PageNumber. bmp

3-5-2 テキストの出力

エンコードする前のテキストデータを、テキストファイルとして出力します。出力するかどうかの選択、出力先、ファイル名などはシンボルの出力と同じルールとします。

文書 1 . doc → 文書 1 _PageNumber. txt

4 種類

音声コード作成ツールとして「ワンクリック版」と、より詳細な設定が可能な「バリアフリー版」の2種類があります。これらは、同じ Word 用テンプレートファイルに納められています。

表 4 音声コード作成ツール機能比較表

	ワンクリック	バリアフリー
シンボルサイズ	M サイズ (17.9mm)	自動可変 (XS~L サイズ) ※ ¹
誤り訂正率	中 (固定)	高/中/低
音質	男性/女性	男性/女性※ ²
音程	中 (固定)	
音量	中 (固定)	
変換対象範囲	全文	<ul style="list-style-type: none"> ・ 選択範囲 ・ 選択されていない場合は全文
複数ページ処理	ページ毎に一つのシンボルを作成※ ³	
操作パネル	なし※ ⁶	表示※ ⁵
画面	<p>[Speechio ツールバー]</p> 	<p>[SP Code Maker ツールバー]</p>  <p>[操作パネル]</p> 

- ※1 SPコードとしてはXS～XLサイズまでの大きさがありますが、製品版音声コード読取機器では物理的にXLサイズの読み取りが不可能であるため、本ソフトウェアでは作成する大きさをLサイズまでとします
- ※2 文単位に設定が可能です
- ※3 ツールバー上のアイコンをクリックすることでコードを作成します
- ※4 ページをまたぐ文は、次のページへ送ります
- ※5 「SP Code Maker ツールバー」上の「詳細設定ボタン」をクリックすることで操作パネルを表示します

4-1 ワンクリック版

- (1) 文書のチェック (`mdlFace.PreCheck`)
 - 文書が開かれているか、文書にテキストが入力されているかを確認します。
 - ※ エラーの場合は、メッセージを表示して終了します
- (2) 変換対象範囲の取得 (`mdlFace.PreCheck`)
 - 本文中の全文を変換対象範囲として取得します。
- (3) 音声コード読取機器の音声出力情報の取得 (`mdlFace.OneClickSet`)
 - 音声出力の設定値として、選択された音質（男性または女性）を取得します。
- (4) エンコードに必要な情報の設定 (`mdlFace.OneClickSet`)
 - シンボルサイズをMサイズに、音程・音量を中に設定します。
- (5) 変換対象の文書があるディレクトリへのパスを取得 (`mdlMain.SpeechioMain`)
- (6) ページの分割位置と、シンボルの貼付位置を取得 (`mdlWordHandle.GetPageInfo`)
- (7) 対象ページ内にあるシンボルを削除 (`mdlWordHandle.ForceDelSPcode`)
- (8) 対象ページ内のテキストデータの取得・音声出力の制御コードを付加
 - ① 対象ページ内のテキストデータを一括で取得
 - ② 文字列処理 (HT、GR、LFを除く制御文字の削除) (`mdlSpAuth.DeleteCtrl`)
 - ③ (3)、(4)で設定した音声出力の設定情報を付加
- (9) エンコードして `bit_string` を作成 (`mdlSPcode.DoEncode`)
 - 以下の設定により `SpeechioEncode` 関数を呼び出し、`bit_string` を作成します。

<code>data_type_string[]</code>	=T
<code>cell_type_string[]</code>	=m
<code>recover_level_string[]</code>	=N
<code>dammy[]</code>	任意のテキスト
<code>data_size</code>	(8)の(テキストデータ+制御コード)のバイト数

data_code[]	(8)の (テキストデータ+制御コード) のデータ
path_name[]	(5)で取得したパス
bit_string[]	

※ SpeechioEncode 関数の戻り値が負である場合は、オーバーしたバイト数をおおよその文字数に変換し、文字数がオーバーした旨のメッセージを表示します。

(10) シンボルの作成 (mdlSPcode. DoSymbol)

以下の設定により SpeechioSymbol 関数を呼び出します。シンボルがクリップボードへ転送されます。

col	M サイズのセル数 (表 1 参照)
row	
symbol_data[]	(9)で作成した bit_string

(11) クリップボード内のシンボルの取得、リサイズと位置の調整

(mdlSPcode. DoSymbol)

(12) ページ毎に繰り返し処理

(7)から(11)の処理を、(6)で取得した全てのページについて繰り返し処理します。

(13) クリップボード内のデータと「元に戻す」の履歴を消去

4-2 バリアフリー版

(1) 文書のチェック (mdlFace. PreCheck)

文書が開かれているか、文書にテキストが入力されているか、選択範囲が本文中のテキストであるか確認します。

※ エラーの場合は、メッセージを表示して終了します。

(2) 変換対象範囲の取得 (mdlFace. PreCheck)

選択範囲が指定されていれば、その選択範囲を変換対象範囲として取得します。何も選択されていなければ、本文中の全ての文を変換対象範囲として取得します。

(3) 音声コード読取機器の音声出力情報の取得

① 新規文書を開く (mdlFace. ShowNewDoc)

② 元文書を一文ごとに複製 (mdlWordHandle. ShowSelectText)

③ 操作パネルの表示

④ ユーザによる音声出力の音質の設定 (frmBarrierFree)

※ 「閉じる」ボタンが押された場合は終了します。

⑤ 文単位に音質 (男性/女性) を取得 (frmBarrierFree. cmdMakeSymbol_Click)

⑥ 誤り訂正率 (高/中/低) を取得 (frmBarrierFree. cmdMakeSymbol_Click)

⑧ シンボル・テキストを出力するかどうか確認 (mdlOutPut. SetFlg)

- ⑨ シンボル・テキストの出力先を取得 (`mdlOutPut.SetOutDirPath`)
- (4) エンコードに必要な情報の設定 (`frmBarrierFree.cmdMakeSymbol_Click`)
音程・音量を中に設定します。
- (5) 変換対象の文書があるディレクトリへのパスを取得 (`mdlMain.SpeechioMain`)
- (6) ページの分割位置と、シンボルの貼付位置を取得
(`mdlWordHandle.GetPageInfo`)
- (7) 対象ページ内にあるシンボルを削除 (`mdlWordHandle.ForceDelSPcode`)
- (8) 対象ページ内のテキストデータの取得・音声出力の制御コードを付加
 - ① 対象ページ内の一文のテキストデータを取得 (`mdlWordHandle.PutCtrlCode`)
 - ② 取得した一文の文字列処理 (HT、CR、LF を除く制御文字の削除)
 - ③ (3)、(4)で設定した音声出力の設定情報を付加 (`mdlWordHandle.PutCode`)
※ 前文と同じ SP 制御コードは付加しません。
- ④ 対象ページの全文について①～③を繰り返し処理し、各文のテキストデータを連結
例) “テスト” というテキストデータに、音質：男、音程：中、音量：中を設定する場合

^V0^H3^P4 テスト

- (9) エンコードして `bit_string` を作成

以下の設定により `SpeechioEncode` 関数を呼び出し、`bit_string` を作成します。シンボルサイズの小さい順からエンコードし、初めて成功したサイズを取得します。

<code>data_type_string[]</code>	=T
<code>cell_type_string[]</code>	=s (小さい順からエンコード)
<code>recover_level_string[]</code>	(3)で取得した誤り訂正率
<code>dammy[]</code>	任意のテキスト
<code>data_size</code>	(8)の (テキストデータ+制御コード) のバイト数
<code>data_code[]</code>	(8)の (テキストデータ+制御コード) のデータ
<code>path_name[]</code>	(5)で取得した文書へのパス
<code>bit_string[]</code>	

※ `SpeechioEncode` 関数の戻り値が負である場合は、オーバーしたバイト数をおおよその文字数に変換し、文字数がオーバーした旨のメッセージを表示します。

- (10) シンボルの作成 (`mdlSPcode.DoSymbol`)

以下の設定により `SpeechioSymbol` 関数を呼び出します。シンボルの画像がクリップボードへ転送されます。

col	(9)で取得したシンボルサイズのセル数
row	
symbol_data[]	(9)で作成した bit_string

- (11) クリップボード内のシンボルの取得、リサイズと位置の調整
(mdlSPcode.DoSymbol)
- (12) シンボルデータ出力 (mdlOutPut.SaveForPrint)
シンボルデータを出力する場合は、クリップボード内のシンボルを BITMAP ファイルとして出力します。出力先は(3)-⑨で指定されたディレクトリとします。
- (13) テキストデータ出力 (mdlOutPut.SaveForPrint)
テキストデータを出力する場合は、(8)で作成したデータの SP 制御コードを取り除き、改行コードを CR+LF に統一して出力します。出力先は(3)-⑨で指定されたディレクトリとします。
- (14) ページ毎に繰り返し処理
(7)から(12)の処理を、(6)で取得した全てのページについて繰り返し処理します。

5 Word マクロの開発

5-1 開発時のファイルの位置

Word マクロの開発を行う場合、フォームやモジュールなどのデータは、全て既定のテンプレートファイルへ保存する。

既存の「Normal.dot」は削除しても構わず、存在しない場合は Word の起動時に自動的に作成されます。また、「Normal.dot」以外のファイル名としておけば、Word の起動時にロードされることはない

※ 既製のテンプレートファイル (Speechio.dot など) を変更したい場合は、上記位置に「Normal.dot」としてコピーすれば変更可能です。

5-2 エディタの起動

Word 上で「ツール」メニューの「マクロ」から、「Visual Basic Editor」を選択します。「プロジェクトエクスプローラ」から、Normal を選択して開発を行う。

Word のバージョンによって VB のバージョンが異なり、使用できない関数などが存在するので注意する。

5-3 マクロの実行

マクロは、Word のツールバーやメニューから実行可能です。新しくツールバーやメニューを作成する場合は、「ツール」メニューの「ユーザ設定」を選択して表示される、「ユーザ設定」ダイアログを使用する。

ツールバーを作成する

「ユーザ設定」ダイアログの「ツールバー」タブを選択し、「新規作成」ボタンを押します。表示されるダイアログで、ツールバー名と保存先を指定します。

メニューを作成する

「ユーザ設定」ダイアログの「コマンド」タブを選択し、「分類」一覧の一番下から「新しいメニュー」を選択します。「コマンド」一覧に「新しいメニュー」が表示されるので、これを既存のメニューバーやツールバーにドラッグ&ドロップする。

次に、作成したマクロをツールバーやメニューに登録します。「ユーザ設定」ダイアログの「コマンド」タブを選択して、「分類」一覧からマクロを選択すると、「コマンド」に利用可能なマクロの一覧が表示されます。このマクロをツールバーやメニューへドラッグ&ドロップして登録する。

「ユーザ設定」ダイアログを表示したまま、登録したマクロを右クリックすると、名前やボタンイメージなどが変更可能となる。

5-4 マクロの配布

作成したテンプレートファイルを配布する際には、Word 上で「スタートアップ」として設定されているフォルダにコピーする必要があります。Word 上からは、以下の手順によって「スタートアップ」に指定されているフォルダを確認できます。

- ① 「ツール」メニューから「オプション」を選択
- ② 「既定のフォルダ」タブをクリックして表示
- ③ ファイルの種類一覧で「スタートアップ」の既定の参照先を確認

「スタートアップ」フォルダは、使用する Word や Windows のバージョンによってパスが異なるため注意が必要です。デフォルトのパスは一般的に以下のようになっています。

Word 97/98

C:\Program Files\Microsoft Office\Office\STARTUP

Word 2000/2002

- ・ Windows 9x の場合

C:\Windows\Application Data\Microsoft\Word\STARTUP

- ・ Windows NT 4.0 の場合

C:\Windows\Profile\<ユーザ名>\Application Data\Microsoft\Word\STARTUP

・ Windows 2000/XP の場合

C:\Documents and Settings\<ユーザ名>\Application Data\Microsoft\Word\STARTUP

また、テンプレートのファイル名は、拡張子を「.dot」とする必要があります。

5-5 音声コード読取機器における「読み」の指定について

音声コード読取機器においてテキストデータを音声出力した際に、目的の「読み」で音声出力されない場合は、その単語の「読み」を指定することが可能です。変換対象のテキストデータに以下の形式で読みを入力します。

(単語:読み)

単語 「読み」を指定したい単語 (**全角**) をあらわします。

読み... 単語の「読み」を**半角カタカナ**で入力します。

※ 括弧とコロンは、半角の () と : を入力します。

例)

株式会社(日本テレソフト:ニホンテリソフト)は

「カブシカ イシャニホンテリソフト」 と音声出力されます。

6 アプリケーション

音声コード読取機器で読み取り可能な SP コードを、Microsoft Word 上で作成するソフトウェアであり、Word 文書内に入力されたテキストデータをもとに SP コード画像を作成し、作成された画像を音声コード読取機器で読み取り可能な大きさと位置になるよう Word 文書内に配置する。この文書を市販のプリンタ (600dpi 以上のレーザープリンタ) で印刷することにより、音声コード読取機器で読み取り可能な SP コード付きの印刷物を作成できる。

SP コード作成機能の他に、市販のタックシールへ SP コードを印刷するためのシール印字機能や、入力されたテキストデータが音声コード読取機器にてどのような読み方で音声出力されるかを PC 上で確認できる読みの確認機能を追加して、様々な SP コード付きの印刷物を効率よく作成できる手段を提供することを目的とした。

ここでは、ソフトウェアの構成と各機能について説明し、ソフトウェアは、Word のアドインファイルと複数の DLL ファイルによって構成されており、これらの内部処理についても説明する。

6-1 ファイル構成

図1に示すように、Word アドインと DLL ファイル、および設定ファイル (ini ファイル) により構成されている。音声合成プログラムに関しては、DLL と辞書ファイルにより構成されている。

DLL ファイルは全てシステムフォルダへ配置し、Word アドインは、Microsoft Word で指定された STRATUP フォルダに配置する必要がある。辞書ファイルと設定ファイルも、このフォルダ内に図1のように配置するものとする。

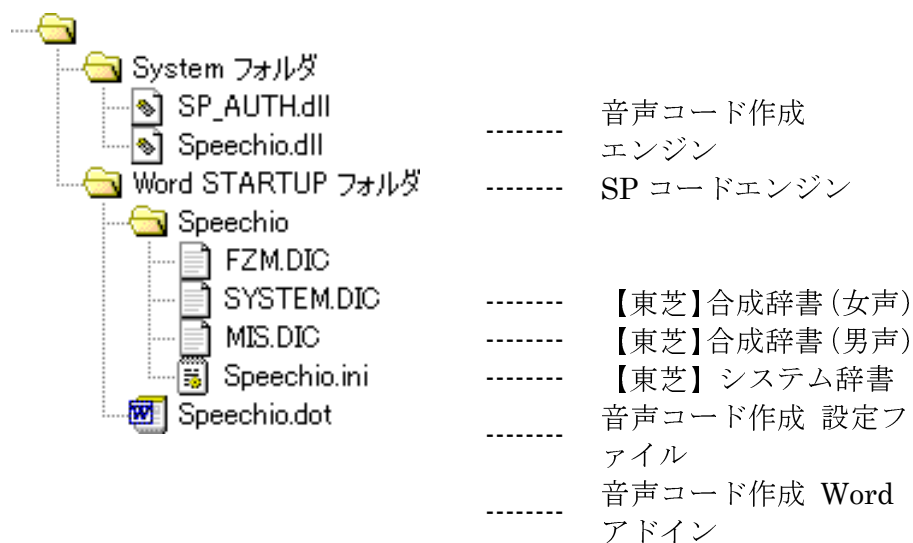


図1 ファイル構成図

6-2 開発環境

Microsoft Windows XP Professional SP2 日本語版	
Microsoft Word 2007 日本語版	----- Word アドイン開発環境
Microsoft Visual C++ 6.0 日本語版	----- DLL 開発環境

6-3 動作環境

本ソフトウェアの動作環境は、基本的に Word VBA のバージョンに依存し、Word 2000 以降であれば動作する。音声合成ライブラリで生成される合成音声波形データが、Windows 95 上で標準では再生されないため、Windows 95、98 は動作対象外とする。動作を確認している環境は以下の通りである。

Microsoft Word 2000~2007 日本語版
Microsoft Windows 2000 Professional/XP Professional~Windows7 日本語版

6-4 目的と機能

基本的に音声コード読取機器で読み取り可能な SP コードを Microsoft Word 上で作成することを目的としている。SP コード作成は、Word 文書のページ内のテキストデータを取得し、1 ページごとに 1 つの SP コードを作成する。音声コード読取機器で複数のコードを読み込んだ時に、どのコードが音声出力されているかを容易に判定できるよう、SP コード作成の際に、データの先頭にページ情報を自動的に挿入することや、他のアプリケーションで SP コードを扱えるよう、SP コードを画像ファイルとして出力することも可能とする。SP コード作成機能は、ボタン一つで SP コードを作成する ワンクリック (男性の声/女性の声) 機能と、テキストデータの文単位に男女の声質を設定して SP コードを作成することが可能な 詳細設定 機能とで構成する。

これに対して シール印字 機能は、SP コードの作成された Word 文書をもとに、市販のタックシールのレイアウトとなるよう SP コードを複製・配置する機能である。用紙サイズや余白、縦と横のシール数などを数値で指定することで、様々な形式のタックシールに対応する。これにより、既存の印刷物や箱型のパッケージなどへ SP コードを貼り付けることができる。

また、音声コード読取機器は、読み込んだ SP コード内のテキストデータを音声合成して音声出力するため、必ずしもコード作成者の意図した読み方で音声出力されるとは限らない。このため、コードを紙へ出力して、音声コード読取機器にて読み方を確認し、もとのテキストデータを修正するという作業が発生する。読みの確認 機能は、Word 上にて読み方の確認作業を行うことのできる音声出力機能であり、コンテンツ作成者の確認作業を軽減して、効率よく SP コードを作成することができる。

各機能では、様々な設定を行うため、次回起動時に同じ設定内容が反映されることが望ましい。環境設定 機能は、設定内容を外部ファイルとして保存する機能であり、ソフトウェアの実行時には、この設定ファイルを読み込んで初期化を行う。

表 1 機能一覧

No.	機能	備考	
1	ワンクリック (男性の声)	全て男声	SP コードの設定は環境 設定にて変更
2	ワンクリック (女性の声)	全て女声	
3	詳細設定	文単位に男女の声質を指定	
4	シール印字	ダイアログを表示して、シールの大きさなどを指定	
5	読みの確認	音声コード読取機器と同じ音声出力（声質は環境設定にて変更）	
6	環境設定	ダイアログ（SP コード作成・読みの確認）を表示して設定	

7 ユーザーインターフェイス（メニューとツールバー）

本ソフトウェアは、Microsoft Word ヘアドインとして組み込み、図 2 に示す「SP コード」メニューとツールバーを表示する。

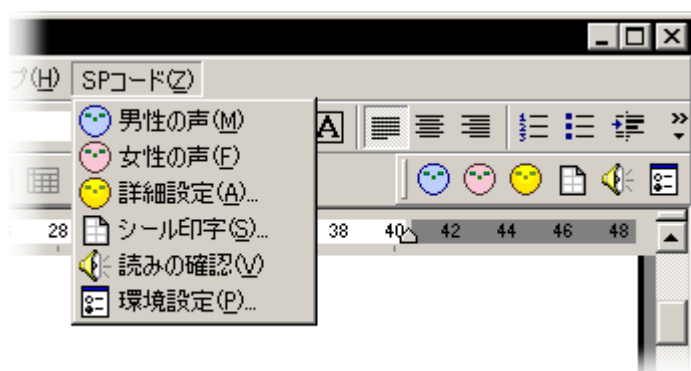


図 2 「SP コード」メニューとツールバー（Microsoft Word 画面）

メニューの項目を選択するか、または、ツールバー上のボタンを押すと、下表に示す機能を実行する。「ワンクリック（男性の声）」・「ワンクリック（女性の声）」・「詳細設定」・「シール印字」・「読みの確認」・「環境設定」のそれぞれに対応した Word アドイン内のプロシージャを呼び出す。

表 2 アイコンと実行される機能

No.	アイコン	機能
1	 (青)	ワンクリック（男性の声）
2	 (赤)	ワンクリック（女性の声）
3	 (黄)	詳細設定
4		シール印字
5		読みの確認
6		環境設定

7-1 機能説明

7-1-1 機能

- ① 指定された声質（男声/女声）で音声出力可能な SP コードを作成する。
- ② Word 文書中のページ内のテキストデータを SP コード化し、1 ページに 1 つの SP コードを作成する。
- ③ 作成する SP コードの設定は、設定ファイルに記録された値を参照する（設定値は「環境設定 SP コード作成」により変更する）。

7-1-2 画面

- ① 正常終了時は作成完了のメッセージ（図 3）を表示する。



図 3 SP コード作成完了メッセージ

7-2 詳細設定

7-2-1 機能

- ① 文単位に指定された声質（男声/女声）で音声出力可能な SP コードを作成する。
- ② Word 文書内の 1 ページに 1 つの SP コードを作成する。
- ③ 作成される SP コードの設定は、設定ファイルに記録された値を参照する（設定値は「環境設定 SP コード作成」により変更する）。

7-2-2 画面

- ① Word 文書内のテキストから文のリストを取得し、詳細設定ダイアログ（図 4）を表示する。
- ② 文のリストで声質を変更したい文を指定し、ボタン操作により「男声」/「女声」/「全て男声」/「全て女声」の設定を可能とする。指定された声質（男/女）をリストの左に表示する（デフォルトは男声）。
- ③ 文のリストでは、「Ctrl+マウスクリック」で複数の文、「Shift+マウスクリック」で連続した複数の文が選択可能。
- ④ 「作成」ボタンを押すと、SP コードの作成処理を開始し、正常終了時は作成完了のメッセージ（図 3）を表示する。
- ⑤ 「キャンセル」ボタンかクローズボックスを押すと、ダイアログを閉じて Word 画面に戻る。

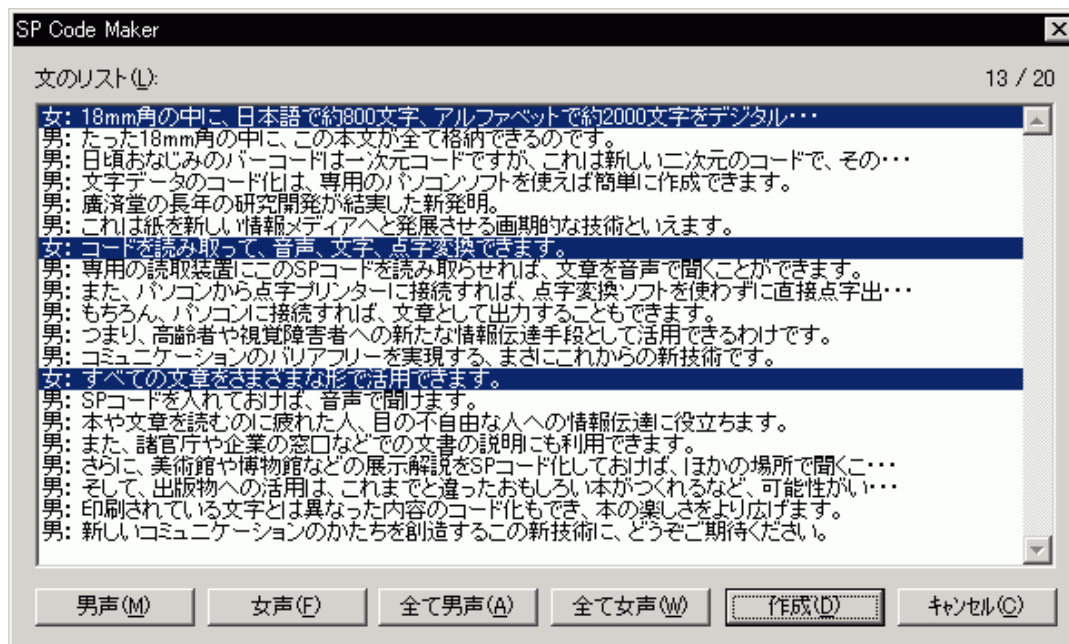


図 4 詳細設定ダイアログ

7-3 読みの確認

7-3-1 機能

- ① Word 文書内で選択されているテキストデータを取得して、音声コード読取機器と同じ音声合成処理によって波形データを生成し、PC のオーディオ出力デバイスから音声として出力する。
- ② 出力する音声の声質の設定は、設定ファイルに記録された値を参照する（設定値は「環境設定 読みの確認」により変更する）。

7-3-2 画面

- ① 読みの確認ダイアログ画面を表示し、音声出力を行う。音声出力中のみダイアログが表示され、全ての出力が終わるとダイアログは自動的に閉じられる。
- ② 「キャンセル」ボタンかクローズボックスを押すと、音声出力を終了し、ダイアログを閉じて Word 画面に戻る。

7-3-3 音声合成ライブラリ

- ① 音声コード読取機器の読みの確認を行うためには、音声コード読取機器内部に組み込まれている音声合成ミドルウェア「TMW39-TTS-A2 ES#35（東芝社製）」と、同等の機能を有し、PC 上で動作するテキスト音声合成プログラムを使用する必要がある。本機能では、音声合成ライブラリ「TVSTX10M2（東芝社製）」（スタティックリンク形式）を使用する。

表 4 音声合成ライブラリの入出力仕様

入力	シフト JIS 全角（ひらがな、カタカナ、漢字、数字、アルファベット、記号）および、半角（カナ、数字、アルファベット、記号）のテキストデータ
出力	22050Hz の 16bit PCM モノラルの波形データ

- ② 音声合成ライブラリには、単語の読み方を登録する辞書登録機能があり、音声コード読取機器内部でも、固定のユーザー辞書（単語数 223 語、「付録 2 音声出力用ユーザー辞書登録内容」参照）の登録処理を行っている。本機能でも音声コード読取機器と単語の読み方を正確に一致させるため、同様のユーザー辞書登録処理を行う。

7-3-4 音声出力用のテキスト整形処理

① 文末の判定

文章を音声合成する際は、抑揚・リズム制御などを適切に行うため、一文単位で音声合成するのが望ましい。音声コード読取機器では、表 5 に示す文字を文末として判定し、テキストデータを文単位に分割して、音声合成処理を行う。本機能でも、同様の処理を行う。

表 5 音声コード読取機器で文末として判定される文字

。）	。）	？）	？）	！）	！）	。
。）	。）	？]	？]	！]	！]	。
。）	。）	？]	？]	！]	！]	？
。）	。）	？}	？}	！}	！}	？
。）	。）	？}	？}	！}	！}	！
。）	。）	？》	？》	！》	！》	！
。）	。）	？」	？」	！」	！」	CR+LF
。）	。）	？」	？」	！」	！」	
。）	。）	？」	？」	！」	！」	
。）	。）	？」	？」	！」	！」	
。）	。）	？」	？」	！」	！」	
。）	。）	？」	？」	！」	！」	
。）	。）	？」	？」	！」	！」	
。）	。）	？」	？」	！」	！」	
。）	。）	？」	？」	！」	！」	
。）	。）	？」	？」	！」	！」	

※1 連続する文末文字は、一つの文末文字として認識される。

※2 CR=0x0d、LF=0x0a をあらかず。

② 機種依存文字の変換

音声合成ライブラリの誤動作を防ぐため、テキストデータ内の全角の機種依存文字（0x84xx~0x87xx）を半角文字に変換する。また、0x84xx~0x87xx までの文字列で表 6 にない文字はスペース（0x20）に変換する。

表 6 機種依存文字の変換表

全角	半角	全角	半角	全角	半角	全角	半角
①	(1)	⑪	(11)	I	I	mm	mm
②	(2)	⑫	(12)	II	II	cm	cm
③	(3)	⑬	(13)	III	III	km	km
④	(4)	⑭	(14)	IV	IV	mg	mg
⑤	(5)	⑮	(15)	V	V	kg	kg
⑥	(6)	⑯	(16)	VI	VI	cc	cc
⑦	(7)	⑰	(17)	VII	VII		
⑧	(8)	⑱	(18)	VIII	VIII		
⑨	(9)	⑲	(19)	IX	IX		
⑩	(10)	⑳	(20)	X	X		

③ 不要データの削除

テキストデータ内の音声出力には関連のない制御コードの削除、および、連続する空白などの削除を行う。

④ 「読みの指定」に関する変換

音声コード読取機器では、テキストデータを音声出力した際に、目的の読み方で音声出力されない単語がある場合は、その単語の読み方を指定することができる。変換対象のテキストデータに以下の形式で読みを入力する。

(単語:ヨミ)

単語 読み方を指定したい単語 (全角)
ヨミ 単語の読み方を半角カタカナで入力
括弧とコロンは、半角の()と:を使用する。

例 株式会社(日本テレソフト:ニホンテレソフト)は
「カブシキガイシャニホンテレソフト」と音声出力される。

単語の後に半角のコロンを、続けて読みを半角カタカナで入力し、単語の始めから読みの終わりまでを半角の丸カッコで囲む。ただし、単語は全角でなくてはならない。

テキストデータ内で上記の形式で読みが指定されている場合は、音声出力用に半角の読み仮名のみを抽出して、音声合成処理を行う。

7-4 環境設定 (SP コード作成/読みの確認)

7-4-1 機能

- ① 「ワンクリック」・「詳細設定」機能にて作成する SP コードの設定と、「読みの確認」機能において音声出力する際の声質の設定ダイログを表示する。
- ② 設定ファイルから設定値を読み込んでダイログを表示し、ダイログ上で変更された値を設定ファイルへ書き込む。

7-4-2 設定内容

ダイアログの「SP コード作成」タブにて設定可能な項目は、表 7 のようになっている。これらの設定値は、「ワンクリック」・「詳細設定」機能にて作成される SP コードに反映される。

表 7 環境設定「SP コード作成」における設定項目と内容

項目	設定内容	初期値	
SP コードの設定	コードサイズ	XS/S/M/L/自動	M
	誤り訂正	強/中/弱	中
	コードの位置	ページの右下/ページの左下/ 偶数ページは右下・奇数ページは左下/ 偶数ページは左下・奇数ページは右下	ページの右下
変換対象範囲		全文/選択範囲のみ	全文
改ページの処理	自動改ページ挿入	ON/OFF	OFF
ページ情報 声の自動挿入	ページ情報	ページ番号/文書名/コメント	挿入しない
	声質	男声/女声/自動（ページ先頭の声質）	—
画像とテキストの出力	ファイル	BMP 画像/テキストデータ	出力しない
	出力先	作成の都度指定/ Word 文書と同じフォルダ/ 常に同じフォルダへ出力	—

① SP コードの設定

- ・「コードサイズ」を「XS/S/M/L/自動」から指定。「自動」を指定すると、ページ内のテキスト量に依存して作成される SP コードの大きさが自動的に変更される。小さいサイズから順にエンコードを行って、データ容量がコードに搭載可能であるかどうかを判定し、初めて搭載可能なサイズとなった時点で SP コードの作成を行う。
- ・「誤り訂正」は、SP コードの読み取りに問題がなければ「中」を推奨している。
- ・「コードの位置」は右下か左下を選択可能とする。見開き印刷や両面印刷の場合を考慮して、偶数/奇数ページで右下と左下に交互に配置できるようにする（「偶数ページは右下・奇数ページは左下」と「偶数ページは左下・奇数ページは右下」とから選択可能とする）。

② 変換対象範囲

- ・「全文」か「選択範囲」のどちらかを選択可能とする。
- ・「選択範囲」を指定し、SP コードの作成時に文章が選択されていない場合は、自動的に全文が変換対象範囲となる。

③ 改ページの処理

- ・「ON」と「OFF」のどちらかを選択可能とする。
- ・「ON」を指定し、ページ内のテキストのデータ量が1つのSPコードに搭載可能なデータ容量を越える場合、文単位にページ範囲を縮小してエンコードを行う。エンコードが成功するまでページ範囲の縮小を繰り返す、縮小したページ範囲の末尾に改ページを挿入する。
- ・「OFF」を指定し、ページ内のテキストのデータ量が1つのSPコードに搭載可能なデータ容量を越える場合は、データ容量がオーバーした旨のメッセージを表示して処理を中止する。「OFF」を指定し、テキストのデータ量がSPコードに納まらない場合は、データ容量がオーバーした旨のメッセージを表示して処理を中止する。

④ ページ情報の自動挿入

- ・「ページ番号/文書名/コメント」の複数を選択可能とし、各項目を“、”で区切ったページ情報データをテキストデータの先頭に挿入する。

〈ページ番号〉ページ、〈文書名〉、〈コメント〉〈改行〉

例3 ページ、SPコード紹介文、SPコードについての説明

- ・「コメント」は20文字以内とする（半角も一文字と数える）。
- ・挿入したページ情報の声質（男声/女声）を設定可能とし、「自動」ではページ先頭の声質と同じ声質を設定する。

⑤ 画像とテキストの出力

- ・作成したSPコードのBitmap画像と、エンコードしたテキストデータをファイルとして出力する。テキストデータには「ページ情報の自動挿入」で挿入したテキストデータも含まれる。
- ・出力先には「作成の都度指定/Word文書と同じフォルダ/常に同じフォルダへ出力」から選択可能とする。「作成の都度指定」を選択した場合は、「フォルダの参照」ダイアログを表示し、保存先を指定可能とする。「Word文書と同じフォルダ」を選択した場合においても、Word文書が保存されていないなど、保存先が特定できない場合は、同様に図13「フォルダの参照」ダイアログを表示する。



図 13 「フォルダの参照」ダイアログ

・出力ファイル名は、以下の形式で自動的に決定する。

〈Word 文書名〉_XXXZ. EXT

XXX : ページ番号

Z : コードサイズ (XS/S/M/L)

EXT : 拡張子 (BMP/TXT)

例 SP コード紹介文_001M. BMP

ダイアログの「読みの確認」タブにて設定可能な項目は、表 8 のようになっている。この設定値は、「読みの確認」機能にて音声出力される声質に反映される。

表 8 環境設定「読みの確認」における設定項目と内容

項目		設定内容	初期値
声質		男声/女声	男声
音声設定	速さ	1~10	5
	高さ	1~10	5
	大きさ	1~10	5
読上げテスト		—	—

「読上げテスト」ボタンを押すと、指定した声質で以下のテキストを例として音声出力し、どのような声質で音声出力されるかを容易に確認できる。

音声コード読取機器 読みの確認機能。この機能は音声コード読取機器と同じ読み方で音声出力を行います。テキストデータが音声コード読取機器でどのように読まれるか、PC上で確認することができる。

8 プログラム構成






8-1 Speechio.dot

Microsoft Word のアドイン形式。主に以下の目的で開発を行っている。

- Word 上からの関数の呼び出し (Word 上へのインターフェイスの実装)
- Word VBA の構文解析機能 (Sentence オブジェクト) や画像処理機能 (Shape オブジェクト) の利用

ツールバーとメニューから呼び出されるプロシージャは、下表のようになっている。

表 9 インターフェイス上から実行されるプロシージャ

アイコン	機能	ファイル名	プロシージャ
 (青)	ワンクリック (男性の声)	mdlSp1Click. bas	SpOneClickMale
 (赤)	ワンクリック (女性の声)	mdlSp1Click. bas	SpOneClickFemale
 (黄)	詳細設定	mdlSpAdvanced. bas	SpAdvanced
	シール印字	mdlSpSeal. bas	SpSealPrinting
	読みの確認	mdlSpVoice. bas	SpReading
	環境設定	mdlSpSetting. bas	SpSetting

8-2 SP_AUTH.dll

DLL 形式の音声コード作成エンジン。主に以下の目的で開発を行っている。

- 文字列処理の速度向上
- ダイアログのスクリーンリーダー (視覚障害者向けの PC 画面読上げソフトウェア) への対応 (VB のフォームはコントロールによってスクリーンリーダーに未対応のものがあるため、C 言語 + Windows API にてダイアログを作成することで対応している。)
- 東芝音声合成ライブラリの呼び出し (スタティックリンクライブラリとして組み込み)
- 音声コード読取機器ソースの移植 (音声コード読取機器では全て C 言語ベースで開発されている)

「Speechio.dot」の内部から呼び出すことを前提としているため、VB から呼び出し可能な形式（_stdcall 呼び出し規約）で開発を行っている。

8-3 Speechio.dll

DLL 形式の SP コード作成エンジン。API 関数などの詳細は、「付録 1 SP Code 作成ソフトウェア 開発者向け仕様書」を参照のこと。また、「Speechio.dot」内では、「mdlSpCode.bas」にて「Speechio.dll」の API 関数が宣言されている。

9 設定ファイルの構成

設定ファイル（Speechio.ini）は、基本的に環境設定ダイログ（「7-4 環境設定」参照）から変更するが、設定ファイルをエディタなどで直接変更することも可能である。設定ファイル内に記述される内容は、表 9 のような構成になっている。

表 9 設定ファイルの構成

セクション名		キー名		値		
項目	意味	項目	意味	値	意味	
EncodeSetting	エンコード設定	DataType	データタイプ	T	日本語テキスト	
				C	コントロールカード	
		CellType	コードサイズ	s	XS サイズ	
				S	S サイズ	
				m	M サイズ	
				M	L サイズ	
				<空>	指定のない場合は自動	
		RecoverLevel	誤り訂正	S	強	
				N	中	
				P	弱	
SpProperty	音声コード作成の設定	Position	コードの位置	0	右下	
				1	左下	
				2	偶数ページは右下	
				3	偶数ページは左下	
		PageBreak	自動改ページ	-1	有り	
				0	無し	
				0	全文	
		EncodeRange	変換対象範囲	1	選択範囲、未選択は全文	
				2	選択範囲のみ	
				0	全文	
		PageInfo	ページ情報	整数	0	(なし)
					1	(ページ番号)
		PageInfoGender	ページ情報の声質		2	(文書名)
4	(コメント)					
合計値						
PageInfoGender	ページ情報の声質		0	男声		
			1	女声		
			2	自動 (ページ先頭と同じ)		
Comment	コメント	文字列				

		SaveText	テキスト出力	-1	有り
				0	無し
		SaveBMP	BMP 画像出力	-1	有り
				0	無し
		SaveDir	保存先	Select	作成の都度指定
				OpusApp	Word 文書と同じ
<パス>	常にこのフォルダへ出力				
Others	その他	LatestDir	前回の保存先	<パス>	前回の<パス>を記録
SpVoice	声質設定	Gender	声質	0	男
				1	女
		Speed	速さ	整数	-20~20
		Pitch	高さ	整数	-100~100
SealPrinting	シール印字設定	PaperWidth	用紙の幅	数値	
				PaperHeight	用紙の高さ
		Orientation	用紙の方向	0	縦
				1	横
		TopMargin	上余白	数値	
		BottomMargin	下余白	数値	
		LeftMargin	左余白	数値	
		RightMargin	右余白	数値	
		Hspace	シールの横の間隔	数値	
		Vspace	シールの縦の間隔	数値	
Col	列数	整数			
Row	行数	整数			
#	コメント行				

設定ファイル内に記述されている文法は、Windows で使用される一般的な ini ファイルと同じで、Windows API の GetProfileString 関数や WriteProfileString 関数などを使用して読み書きを行う。

10 Word アドインの開発について

ここでは、Word に組み込んだエンコーダープログラムの環境、起動、実行について説明する。

10-1-1 開発時のファイルの位置

Microsoft Word VBA にて開発を行う場合、フォームやモジュールなどのデータは、全て既定のアドインファイルである「Normal.dot」へ保存する。今回の開発環境である Word 2000/Windows 2000 の場合は、以下の場所に「Normal.dot」が存在する。

C:\Documents and Settings\<ユーザ名>\Application Data\Microsoft\Templates

既存の「Normal.dot」は削除しても構わず、存在しない場合は Word の起動時に自動的に作成される。また、上記の場所にアドインを配置しても「Normal.dot」以外のファイル名としておけば、Word の起動時にロードされることはない。

既製のアドイン（Speechio.dot など）を変更したい場合は、上記位置に「Normal.dot」としてコピーすれば編集可能となる。

10-1-2 エディタの起動

Microsoft Word 上にて「ツール」メニューの「マクロ」から、「Visual Basic Editor」を選択する。画面左側に表示される「プロジェクトエクスプローラ」から、「Normal」を選択して開発を行う。Word のバージョンによって VB のバージョンも異なり、使用できない関数などが存在するため注意が必要である。

10-1-3 マクロの実行

作成したマクロは、Word 上のツールバーやメニューから実行可能である。新しくツールバーやメニューを作成する場合は、「ツール」メニューの「ユーザー設定」を選択して表示される、「ユーザー設定」ダイログを使用する。

ツールバーを作成するには、「ユーザー設定」ダイログの「ツールバー」タブを選択して、「新規作成」ボタンを押し、表示されるダイログでツールバー名と保存先を指定する。また、メニューを作成するには、「ユーザー設定」ダイログの「コマンド」タブを選択して、「分類」一覧の一番下にある「新しいメニュー」を選択する。「コマンド」一覧に「新しいメニュー」が表示されるので、これを既存のメニューバーやツールバー上にドラッグ&ドロップする。

次に、作成したツールバーやメニュー内にマクロを登録する。「ユーザ設定」ダイログの「コマンド」タブを選択し、「分類」一覧からマクロを選択すると、「コマンド」に利用可能なマクロの一覧が表示される。このマクロをツールバーやメニューへドラッグ&ドロップして、目的の順番となるよう配置する。

また、「ユーザ設定」ダイアログを表示したまま、登録したマクロを右クリックすると、名前やボタンイメージなどが変更可能となる。

10-1-4 マクロの設定

作成したマクロを設定する際には、Word 上で「スタートアップ」として設定されているフォルダにコピーする必要がある。Word 上からは、以下の手順によって「スタートアップ」に指定されているフォルダを確認でき、このフォルダ内にマクロを配置する。

- ① 「ツール」メニューから「オプション」を選択
- ② 「既定のフォルダ」タブをクリックして表示
- ③ ファイルの種類一覧で「スタートアップ」の既定の参照先を確認

「スタートアップ」フォルダは、使用する Word や Windows のバージョンによってパスが異なるため、注意が必要である。デフォルトのパスは一般的に以下のようになっている。

Word 97/98

C:\Program Files\Microsoft Office\Office\STARTUP

Word 2000/2002

・ Windows 9x の場合

C:\Windows\Application Data\Microsoft\Word\STARTUP

・ Windows NT 4.0 の場合

C:\Windows\Profile<ユーザ名>\Application Data\Microsoft\Word\STARTUP

・ Windows 2000/XP の場合

C:\Documents and Settings<ユーザ名>\Application Data\Microsoft\Word\STARTUP

また、アドインのファイル名は任意であるが、拡張子を必ず「.dot」とする必要がある。

10-1-5 アドインの保護

本ソフトウェア中のアドイン「Speechio.dot」では、内容を表示するためにパスワード入力が必要となるよう保護を行っている。以下の手順で、内容の表示とパスワードの変更が可能である。

- ① 「Speechio.dot」を既定のアドイン「Normal.dot」としてコピーする。
- ② Word を起動し、「ツール」メニューの「マクロ」から「Visual Basic Editor」を選択する。
- ③ 「プロジェクトエクスプローラ」から「SP_CODE_MAKER (Normal)」を選択する。
- ④ 表示されるパスワード入力のダイアログで、「r2d2mateC」と入力すると、プロジェクトが編集可能となる。パスワードを変更するためには、「Visual Basic Editor」上の「ツール」メニューの「SP_CODE_MAKER のプロパティ」を選択し、表示されるプロジェクトプロパティダイアログにある「保護」タブにて変更する。

第3章 デコード（データ<音声>出力部）プログラム

仕様説明

以下、断りがない限り、

文字コード：シフト JIS コード（半角カナ、半角 ASCII 文字を含む）

SP 制御コード：音声コード読み取り機器からの出力用制御コード

SP format：SP 制御コード + テキスト

とする。

1-1 音声出力

音声出力のため、音声エンジンに送信する音声データへ変換する。

1-1-1 処理の流れ

音声データ変換モジュールの処理流れを図 1-1-1 に示す。

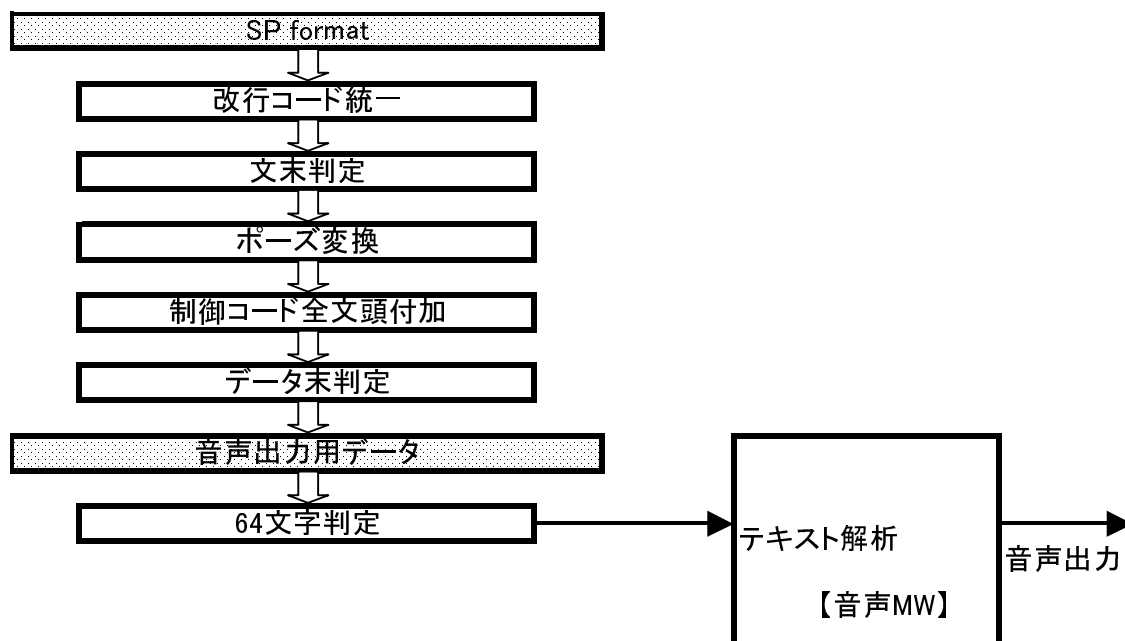


図 1-1-1 音声データ変換モジュールの処理流れ

1-1-2 音声出力用データ作成

1-1-2-1 改行コード統一

データ作成をした OS、ソフトウェアによって改行コードに違いがある。
そこで改行コードを CR(0x0d)+LF(0x0a)に統一する。

改行コード : CR+LF

1-1-2-2 文末判定

文単位で音声出力を行うため、文末を判定し NULL(0x00)を挿入し、文単位に区切る。

文末文字を表 1-1-1 に示す。

表 1-1-1 文末文字

。)	。)	?)	?)	!)	!)	。
。]	。]	?]	?]	!]]	!]]	。
。]	。]	?]	?]	!]]	!]]	?
。}	。}	?}	?}	!] }	!] }	?
。>	。>	?>	?>	!] >	!] >	!
。》	。》	?》	?》	!] 》	!] 》	!
。↓	。↓	?↓	?↓	!] ↓	!] ↓	CR+LF
。』	。』	?』	?』	!] 』	!] 』	
。】	。】	?】	?】	!] 】	!] 】	
。)	。)	?)	?)	!)	!)	
。]	。]	?]	?]	!]]	!]]	
。}	。}	?}	?}	!] }	!] }	
。↓	。↓	?↓	?↓	!] ↓	!] ↓	
。 ”	。 ”	? ”	? ”	!] ”	!] ”	
。 ’	。 ’	? ’	? ’	!] ’	!] ’	

ここで、文末文字は連続しないものとする。

(1) 直前に NULL が挿入されている場合は、文末文字を削除し、NULL は挿入しない。

(2) 直前に NULL+スペースが挿入されている場合は、スペース、文末文字を削除し、NULL は挿入しない。

1-1-2-3 ポーズ変換

音声出力の際、改行、TAB で一定の間隔を空けるため、間隔の空く記号に変換する。

(1) 改行

改行[CR+LF]は、"。(0x8142)に変換し、NULL を挿入する。

(2) TAB

TAB(0x09)は、"。(0x8141)に変換する。

(2-1) 直前に NULL が挿入されている場合は、[TAB]を削除する。

(2-2) 直前が"、"の場合は、[TAB]を削除する。

1-1-2-4 機種依存文字

文章中に機種依存文字が入っていると、音声エンジンの仕様により、予期せぬ動作を示す場合がある。

そこで、機種依存文字 (0x84xx~0x87xx) を半角文字に変換する。

0x84xx~0x87xx までの文字で表 1-1-2 にない文字はスペース(0x20)に変換する。

表 1-1-2 機種依存文字変換表

全角	半角	全角	半角	全角	半角	全角	半角
①	(1)	⑪	(11)	I	I	mm	mm
②	(2)	⑫	(12)	II	II	cm	cm
③	(3)	⑬	(13)	III	III	km	km
④	(4)	⑭	(14)	IV	IV	mg	mg
⑤	(5)	⑮	(15)	V	V	kg	kg
⑥	(6)	⑯	(16)	VI	VI	cc	cc
⑦	(7)	⑰	(17)	VII	VII		
⑧	(8)	⑱	(18)	VIII	VIII		
⑨	(9)	⑲	(19)	IX	IX		
⑩	(10)	⑳	(20)	X	X		

1-1-2-5 読み仮名指定

SP format には、音声出力時の読み仮名の情報が付加されている場合がある。

そこで、単語データを削除する。

例えば、(表記:ヒョウキ)と読み仮名の指定がされている場合、“ヒョウキ”と発音する。

音声出力では読み仮名部分“ヒョウキ”のみ必要なので、単語データを削除し、読み仮名部分を取り出す。

1-1-2-6 SP 制御コード全文頭付加

文章の先頭に、前文までの SP 制御コードの設定を更新し付加する。

これは、SP 制御コードにより設定される文章を飛ばしても、正しい設定で再生する為に行う。

例) 声質設定の制御コードを[男声]:男性の声、[女声]:女性の声とする。

文章 1 : [男声]僕は男です。

文章 2 : 女の子が好きです。

文章 3 : [女声]私は女です。

この時、文章 1-2-3 と再生された時は問題ないが、文章 3 まで再生後に文章 2 に戻った時に、文章 3 を再生した時点で音声エンジンは女性の声に設定されているので、文章 2 は女性の声で再生されることになる。

これを全文章の頭に前文までの SP 制御コードを付加することにより、

文章 1 : [男声]僕は男です。

文章 2 : [男声]女の子が好きです。

文章 3 : [女声]私は女です。

文章 3-2 と再生されても、正しい設定で再生できる。

1-1-2-7 データ末判定

音声データ末に NULL(0x00),NULL(0x00)を挿入し、データ末を区切る。

1-1-3 データ出力

音声出力用データを音声エンジンに介してスピーカー (PC) から出力する。

音声エンジンへのデータ送信は OS に準拠された仕様で送られる。

1-2 テキスト出力

テキスト出力のため、音声出力用制御コード等、テキスト以外の情報を削除する。

1-2-1 処理の流れ

テキストデータ変換モジュールの処理流れを図 1-3-1 に示す。

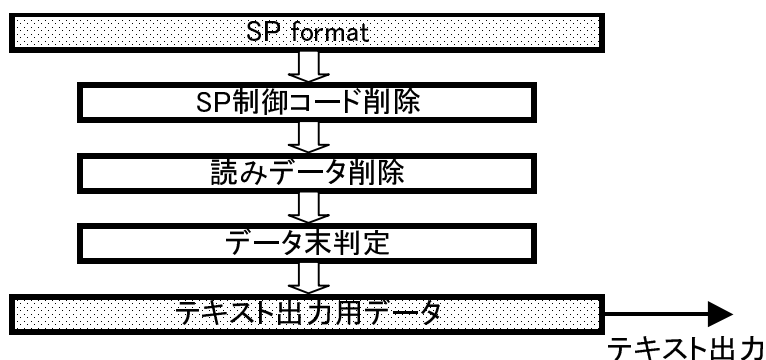


図 1-3-1 テキストデータ変換モジュールの処理流れ

1-2-2 テキスト出力用データ作成

1-2-2-1 SP 制御コード削除

SP format にはテキスト以外の情報の SP 制御コードが付加されている。

そこで CR[0x0d]、LF[0x0a]、TAB[0x09]、NULL[0x00]以外の制御コードを削除する。

1-2-2-2 読み仮名指定

SP format には、音声出力時の読み仮名の情報が付加されている場合がある。

そこで読み仮名データを削除する。

例えば、(表記:ヒョウキ)と読み仮名の指定がされている場合、“ヒョウキ”と発音する。

テキスト出力では単語部分“表記”のみ必要なので、読み仮名データを削除し、単語部分を取り出す。

1-2-2-3 データ末判定

テキストデータ末に NULL(0x00),NULL(0x00)を挿入し、データ末を区切る。

1-2-3 データ出力

テキスト出力用データを画面表示へと出力する。

出力されたデータを文字サイズ変更、表示カラー変更設定により文字表示変更させる。

1-3 関数一覧

uchar : typedef unsigned char

SP_PRINTER_CONTROL 構造体 : SP player の項を参照。

BRLCODE 構造体 :

```
typedef struct {
```

```
    int index;           /* 英数記号もしくは、かな漢字の開始地点 */
```

```
    uchar* code;        /* 中間言語を取得したバッファのアドレス */
```

```
    int length;         /* 英数記号もしくは、かな漢字の長さ */
```

```
    int flg;            /* 英数記号、かな漢字判別フラグ */
```

```
} BRLCODE;
```

```
#define FLG_CLEAR      0      /* デフォルトのフラグ */
```

```
#define FLG_ALNUM      1      /* 英数記号のフラグ */
```

```
#define FLG_NOTALNUM   2      /* かな漢字のフラグ */
```

1-3-1 音声出力

関数名 : int SPtoVoice(uchar* voice_data,uchar* sp_data,int sp_size)

引き数 : (入力) sp_data:デコード直後の SP format

 sp_size:sp_data のサイズ

 (出力) voice_data:受け取りバッファ

戻り値 : voice_data のサイズ

機能 : 全文頭に制御コードをつけた音声エンジン送信用データを作成。

注意 : 受け取りバッファサイズは入力サイズの 2 倍以上確保。

1-3-2 テキスト出力（画面表示）

関数名 : int SPtoText(uchar* text_data,uchar* sp_data,int sp_size)

引き数 : (入力) sp_data:デコード直後の SP format

sp_size:sp_data のサイズ

(出力) text_data:受け取りバッファ

戻り値 : text_data のサイズ

機能 : 音声出力用制御コードを削除し、テキスト出力用データを作成。

2 デコードプログラム生成

【目的】

it 機上の SP コードデモアプリケーションにおける、スキャン画像からの SP コード抽出・デコードモジュールの設計仕様について述べるものである。

【概要】

SP コードデコーダでは、以下の処理を行う。

- ・ スキャナ画像からの SP コード部分抽出
- ・ 抽出した SP コードデータからのテキスト抽出

本モジュールは、SP コード読み取り機器 スピーチオのソースを Windows 上にポーティングすることにより実現している。

したがって、基本アルゴリズムはスピーチオ実機と同等のものである。

2-1 動作環境・開発環境

動作環境

本モジュールは、実装ハード (it 機) に依存する部分がないため、汎用 Windows 用 DLL として実装する。

したがって、本モジュールの動作環境は以下の通りとなる。

Windows 2000 Professional/Server 以降

開発環境

本モジュールは、以下の環境で開発する。

Microsoft Visual C++ V6.0 SP5

2-2 I/F 仕様

本モジュールは、以下の 2 つの関数をエクスポートする。

- **Generator**
画像データからの SP コード取り出し
- **Decoder**
SP コードデータからのテキスト抽出

2-2-1 画像データからの SP コード取り出し

形式

```
int Generator(unsigned char *img_data,  
              int *row, int *col, unsigned char **bit_string)
```

引数

<u>名前</u>	<u>I/O</u>	<u>説明</u>
img_data	in	画像データ。1024x1024 ピクセルの 2 値データ
row	out	副走査方向の SP コードユニット数
col	out	主走査方向の SP コードユニット数
bit_string	out	SP コードデータビット文字列。"0", "1"で構成される。

戻り値

0	成功
負値	失敗

概要

引数で指定された画像から、SP コード部分を抜き出す。
bit_string で返されるメモリは開放の必要なし。
スレッドセーフではないので注意が必要。

2-2-2 SP コードデータからのテキスト抽出

形式

```
int Generator(int row, int col, unsigned char *bit_string,  
             unsigned char **data)
```

引数

<u>名前</u>	<u>I/O</u>	<u>説明</u>
bit_string	in	SP コードデータビット文字列。”0”, “1”で構成される。
row	in	副走査方向の SP コードユニット数
col	in	主走査方向の SP コードユニット数
data	out	抽出テキスト

戻り値

0	成功
負値	失敗

概要

引数で指定された SP コードデータからテキストを抽出する。
data で返されるメモリは開放の必要なし。
スレッドセーフではないので注意が必要。

3 SP コード抽出処理詳細

SP コード抽出を行う上で、対象ハードの違いにより SP コード抽出処理は単純なポーティングでは動作しないことがわかっている。

そのため、元ソースを解析した結果を元に以下の通り、SP コード抽出処理のポーティングを実施した。

以降、元ソースの解析結果を記す。

3-1 ReadSymbol()関数

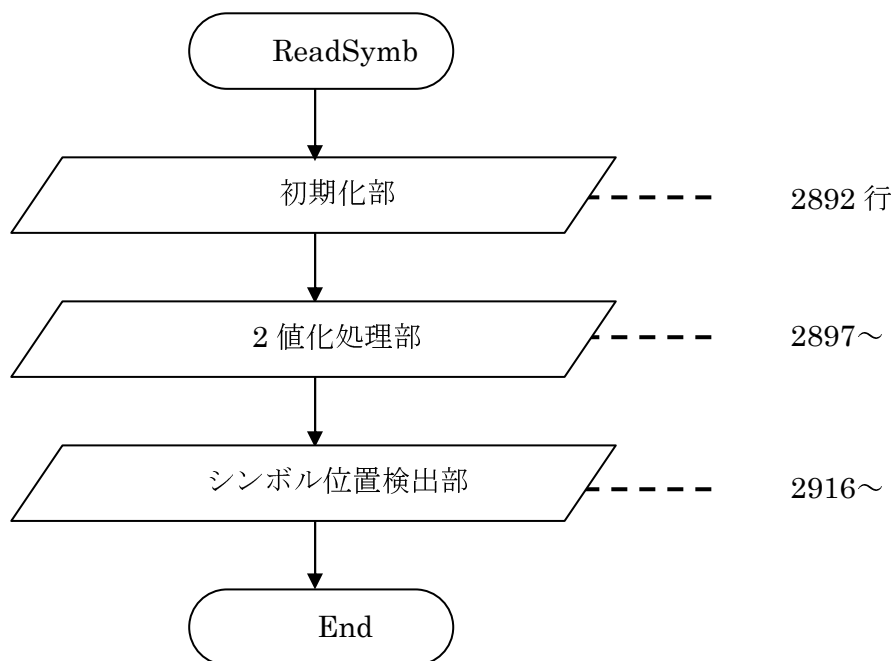
画像データから、2次元コードビット列データを作成する。

【引数】

gray_buf	: 8ビットグレイスケール 画像データ (1024×1024ピクセル)
xsize	: シンボルの横のバイト数
ysize	: シンボルの縦のバイト数
row	: シンボルの横のセル数

col : シンボルの縦のセル数
 data : エンコード用ビット列データを返すバッファ
 【戻り値】
 SUCCESS 成功
 FAILURE 失敗

3-2 全体の流れ



3-3 各部詳細

以下にそれぞれの関数一覧を記す。

(1)初期化部

関数	定義位置	説明
InitSymbolReader	2613~2625 行	初期化する。

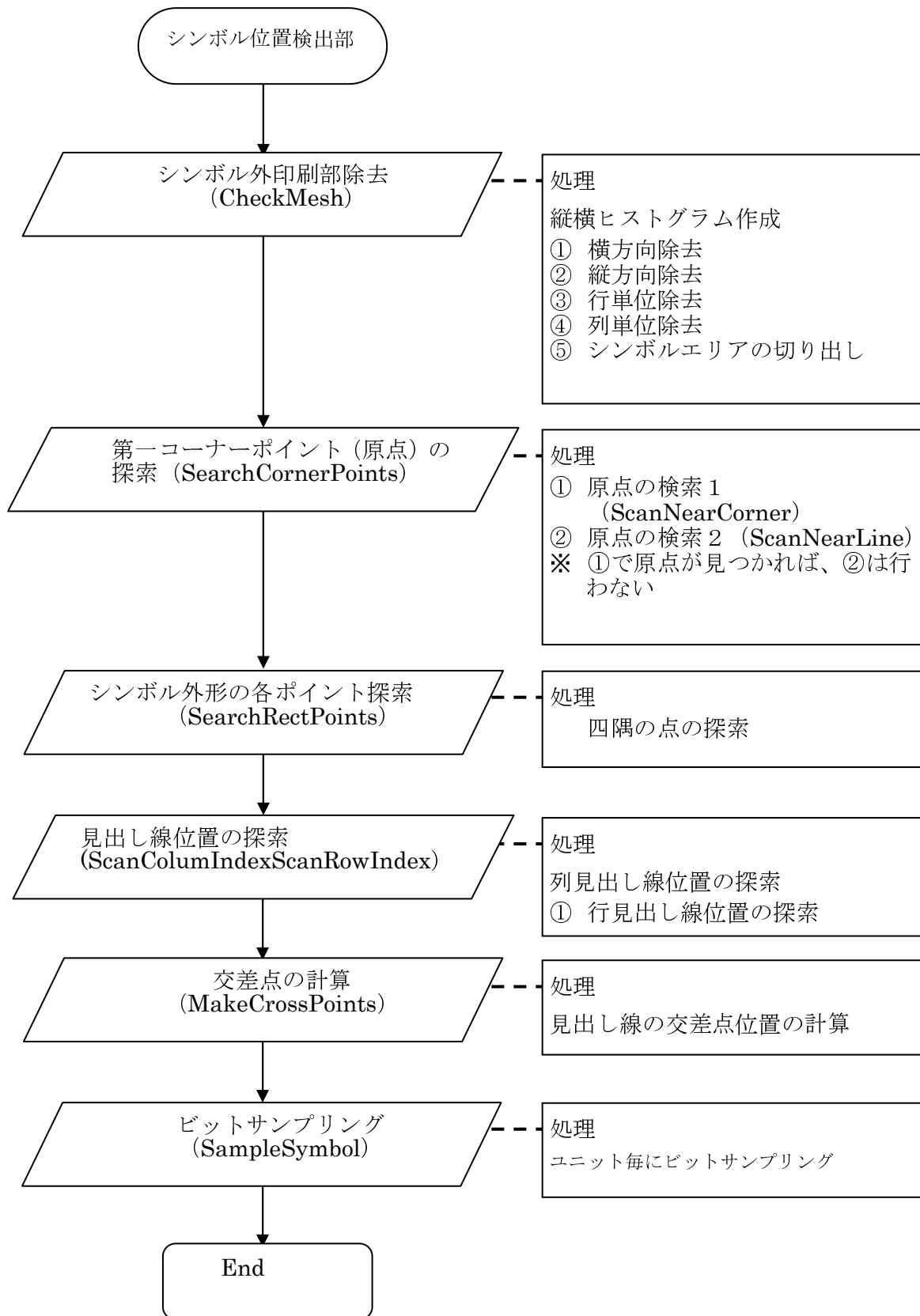
(2)2 値化処理部

関数関数	定義位置	説明
ModeBinarization	436～568 行	モード法により閾値を決定する。
SetupValidRect	423～434 行	エッジを強調する。
Binarization2	702～743 行	2 値化処理を行う。
KillBinaryNoise	816～866 行	ノイズを除去する。

(3)シンボル位置検出部

関数関数	定義位置	説明
CheckMesh	2629～2880 行	シンボル外印刷部を除去する。
SearchCornerPoints	2499～2523 行	第一コーナーポイントを探索する。
SearchRectPoints	1723～1909 行	シンボル外形の各ポイントを探索する。
ScanColumIndex	2448～2471 行	見出し線位置（列）を探索する。
ScanRowIndex	2473～2496 行	見出し線位置（行）を探索する。
MakeCrossPoints	1949～1970 行	見出し線が交差する点を計算する。
SampleSymbol	2547～2609 行	ビットをサンプリングする。

シンボル位置検出の流れ



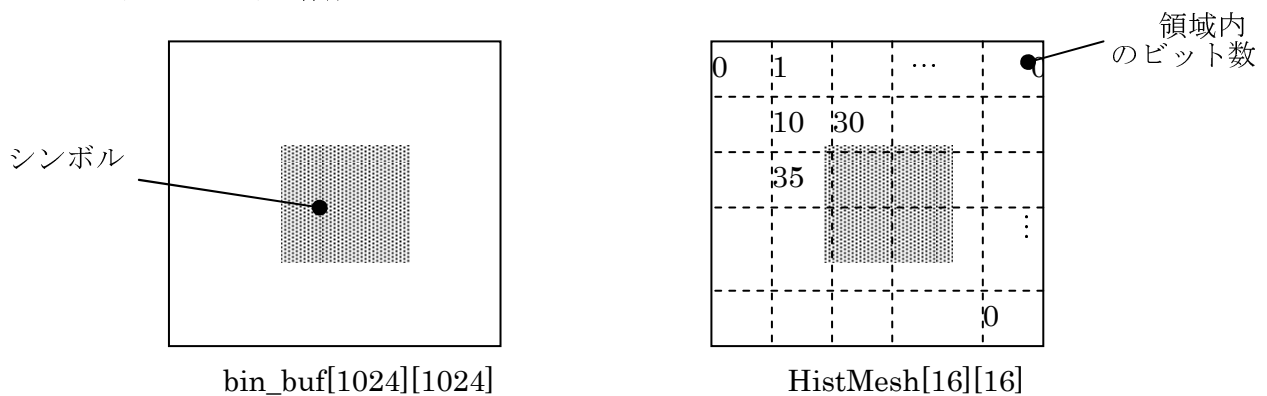
4 各処理の詳細

4-1 シンボル外印刷部除去

2値画像データの水平垂直の分布状態を調査して、2次元コード（シンボル）の存在する矩形エリアを特定する。

① 縦横ヒストグラムの作成

2値画像データ（bin_buf）を16×16の領域に分割し、ビットの度数分布（HistMesh）を作成する。

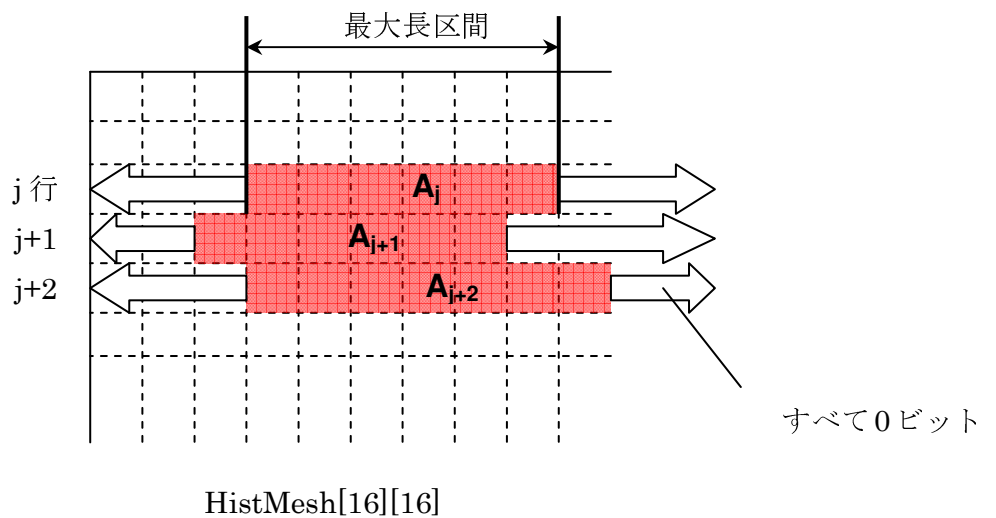


② 横方向除去

横方向に、ビット数が2以上連続して存在する最大長の区間をAとする。

HistMesh各行で、区間Aを除く領域を全て0ビットとする。

0ビットとしたHistMesh領域内のbin_bufも同様に0ビットとする。



③ 縦方向除去

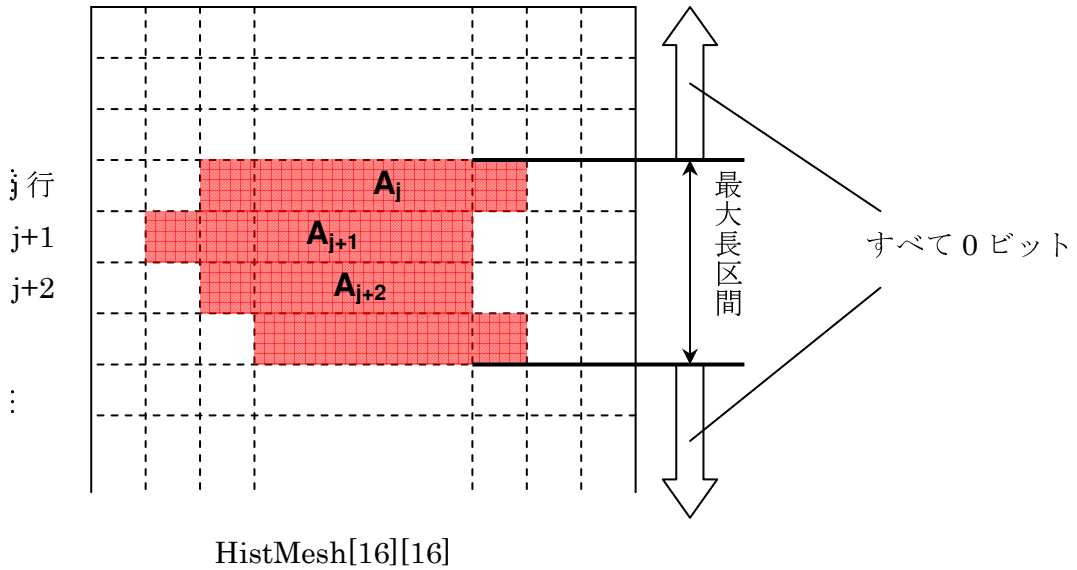
横方向除去と同様の手順で、縦方向に 0 ビットとする。

④ 行単位除去

行単位で、ビットのある領域の最大長区間を探す。

最大長区間を除く領域を全て 0 ビットとする。

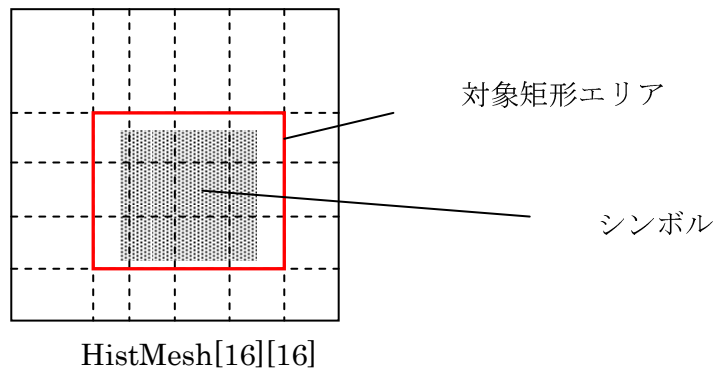
0 ビットとした HistMesh 領域内の bin_buf も同様に 0 ビットとする。



⑤ 列単位除去

行単位除去と同様の手順で、列単位に 0 ビットとする。

②～⑤の処理後、対象矩形エリア以外全て 0 ビットとなる。



⑥ シンボルエリアの切り出し

対象矩形エリア内の、横方向 (x) 及び縦方向 (y) の最大/最小位置を探索する。

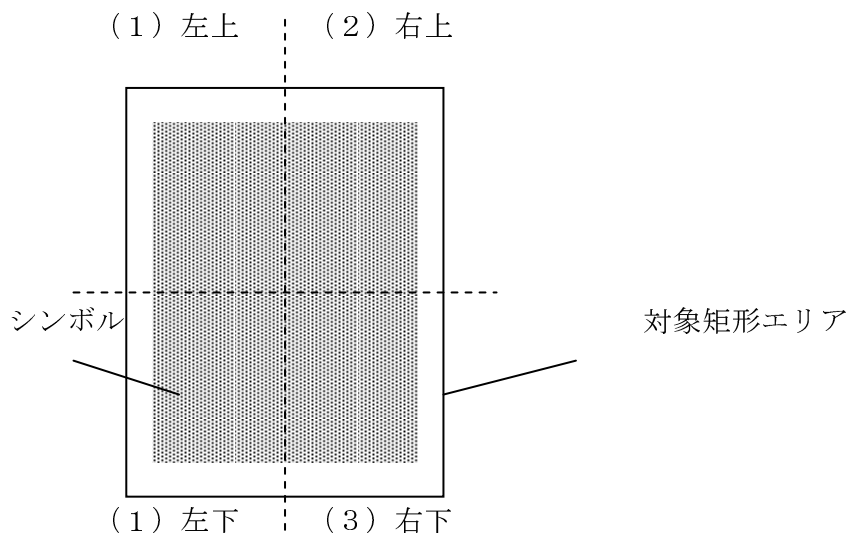
4-2 第一コーナーポイントの探索

コーナーポイント (P0、P1、P2、P3) を探索し、第一コーナーポイント (原点 Pt0) を検索する。

① 原点の検索 1

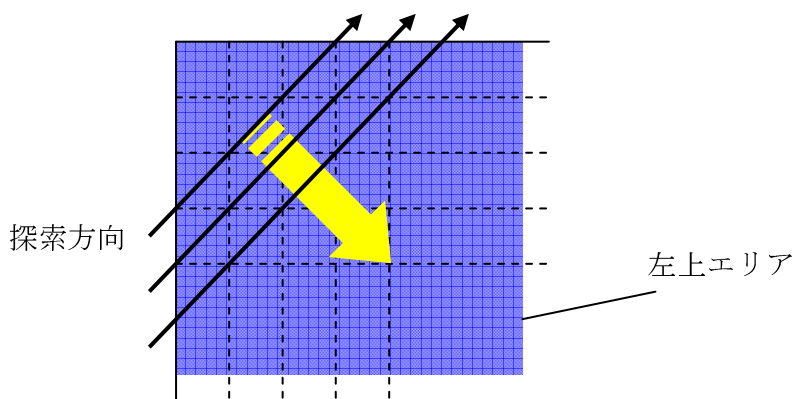
【探索エリアとコーナーポイントの探索】

対象矩形エリアから、左上コーナー⇒右上コーナー⇒右下コーナー⇒左下コーナーの順で、コーナーポイントを探索する。



(1) 左上エリア

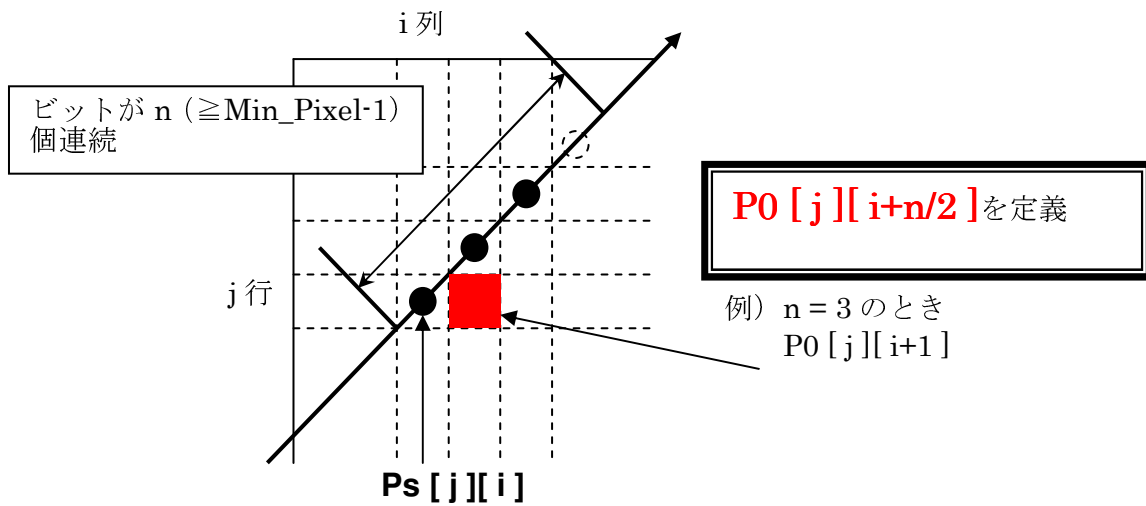
左上コーナーから対角方向に、下図のように斜めに探索する。



探索方向にビット数が (Min_Pixel - 1) 以上連続して存在する場合に、**左上コーナーポイント (P0)** を定義する。このとき、連続開始点を $Ps[j][i]$ とする。

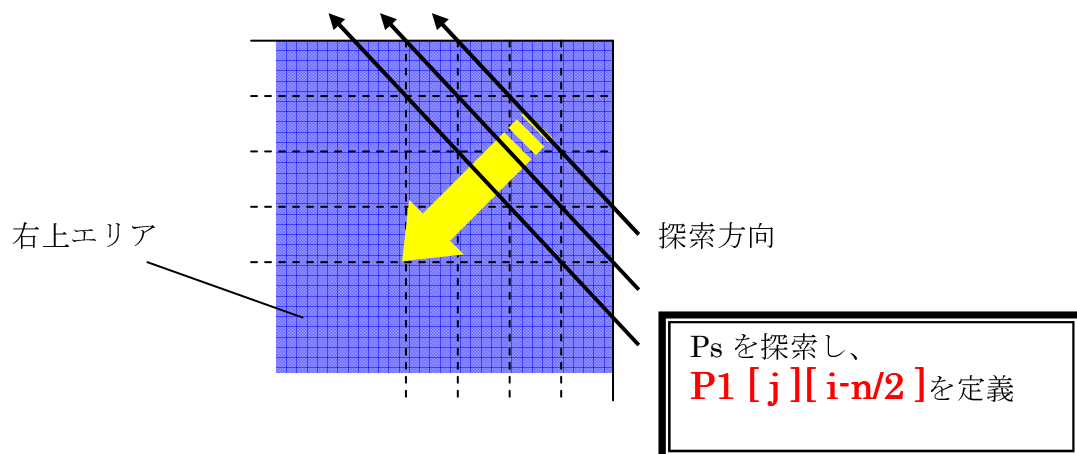
ここで、Min_Pixel は、1 ドットあたりの最小ピクセル数を示す。

※本バージョンでは、Min_Pixel = 4 である。



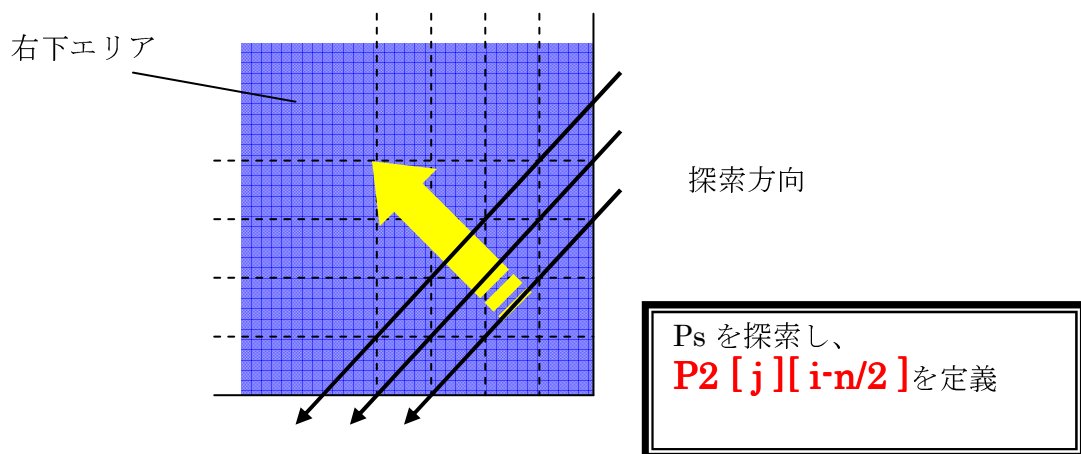
(2) 右上エリア

右上コーナーから対角に向かって、下図のように斜めに探索し、**右上コーナーポイント (P1)** を定義する。



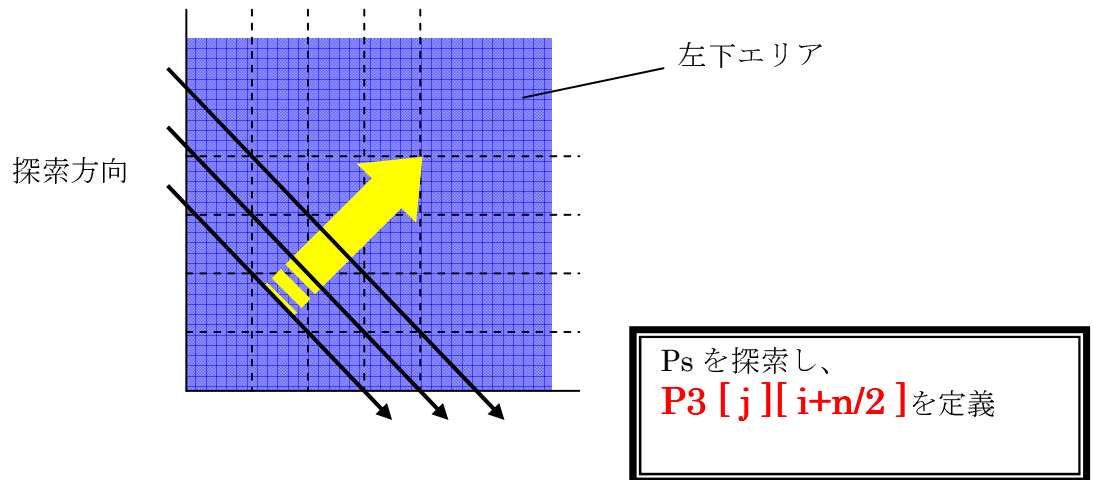
(3) 右下エリア

右下コーナーから対角に向かって、下図のように斜めに探索し、**右下コーナーポイント (P2)** を定義する。



(4) 左下エリア

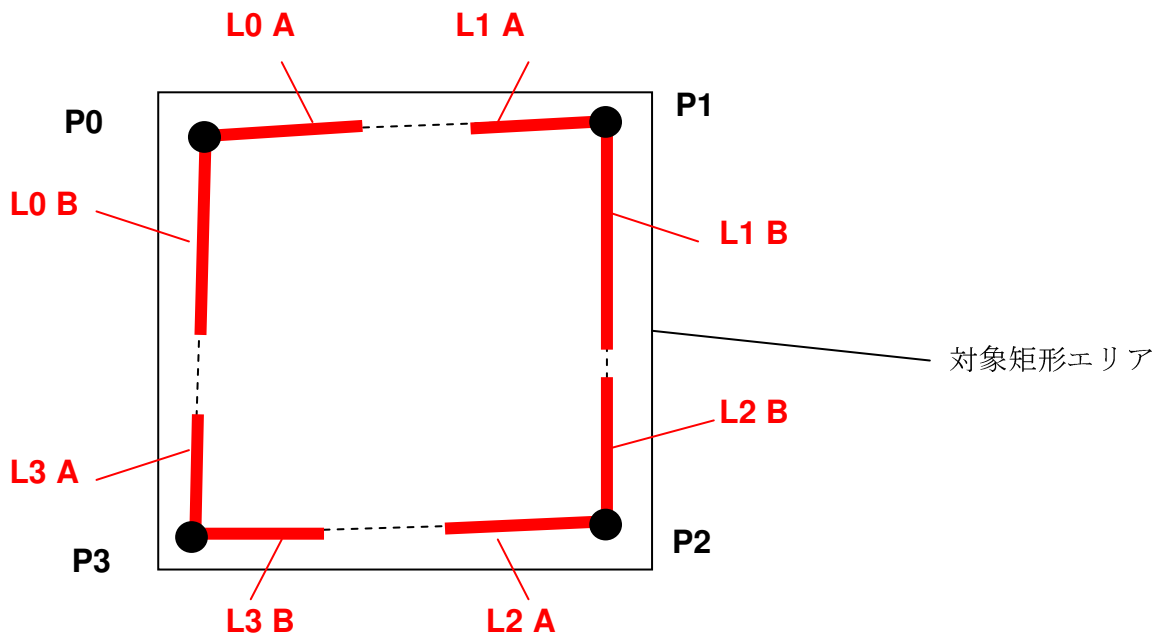
左下コーナーから対角方向に、下図のように斜めに探索し、左下コーナーポイント (P3) を定義する。



【原点の決定】

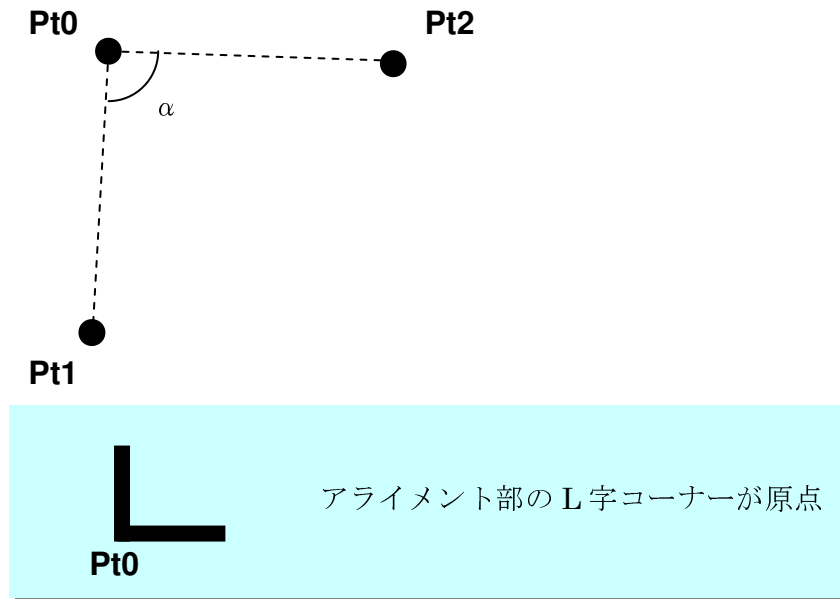
各コーナーポイントで、L を定義する。

Bresenham の線分発生アルゴリズムを用いて、コーナーポイント P0~P3 を結ぶ線分上の点 (座標) を生成し、その点上で、コーナーポイントからビットが連続して存在する長さを L とする。



$LA + LB$ が最大であるコーナーポイントを**原点 (Pt0)** とし、原点の反時計方向の点を Pt1、時計方向の点を Pt2 とする。

ここで、 $\angle Pt1Pt0Pt2$ の成す角度 α は、 $85^\circ < \alpha < 95^\circ$ でなければならない。

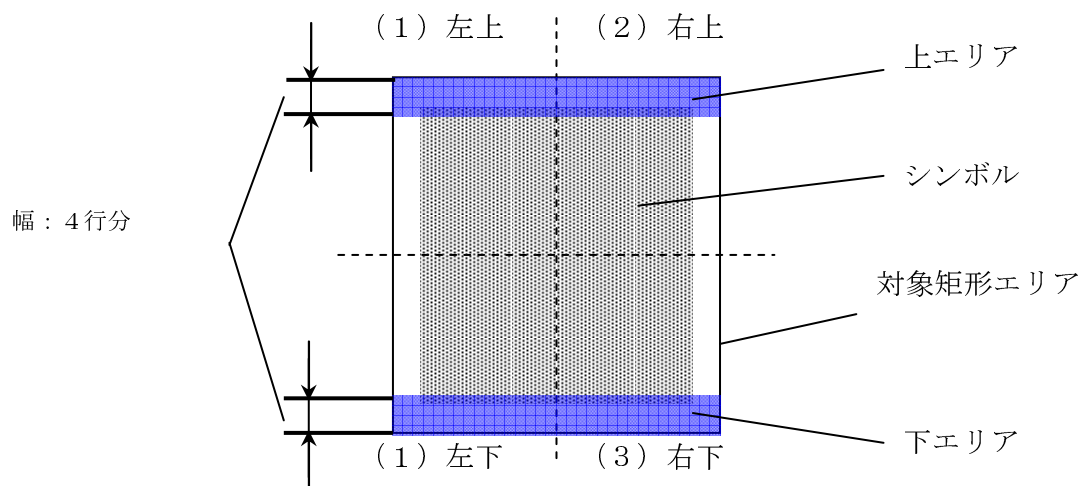


原点の検索 2

【探索エリアとコーナーポイント探索】

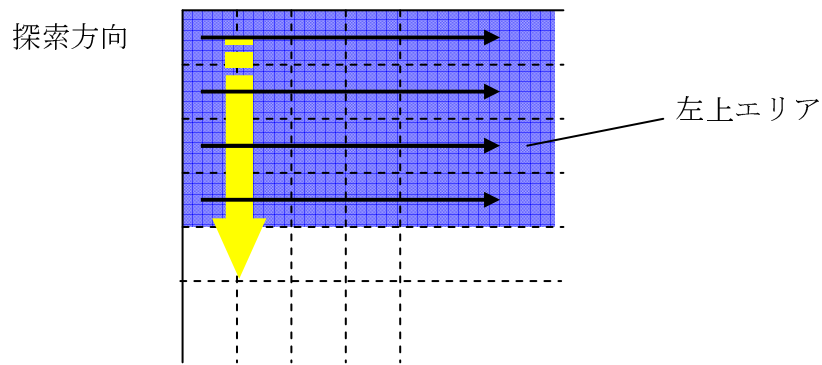
対象矩形エリアの、上下エリア（下図参照）から探索する。

（上下エリアで見つからなければ、左右エリアから探索する）



(1) 左上エリア

左上コーナーから、下図のように横方向に探索する。



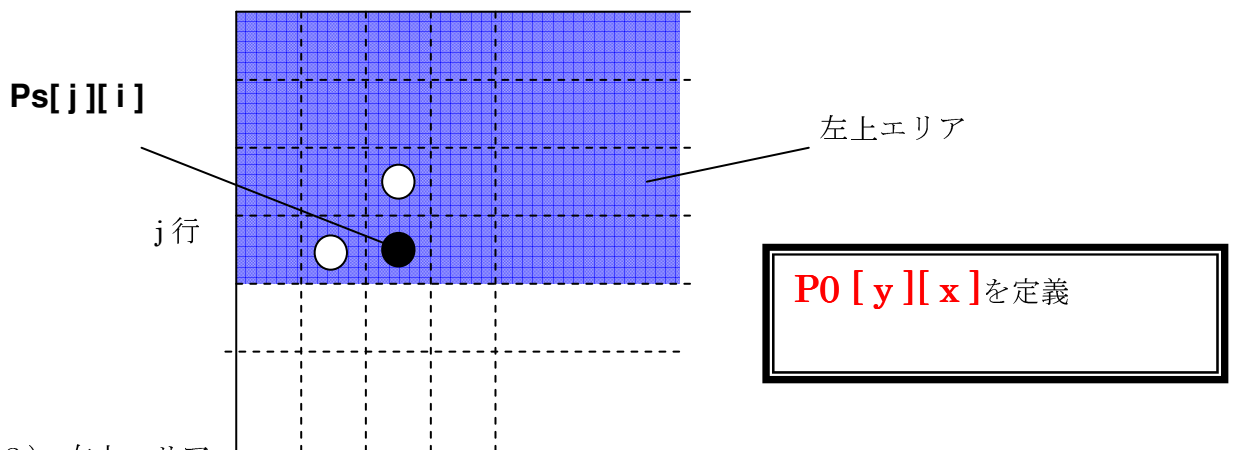
$i=0$ から探索し、ビットがあれば次の行を探索する。探索エリア内で全行探索し、最後にビットが存在した点を $Ps[j][i]$ とする。

左上コーナーポイント ($P0$) の横方向位置、縦方向位置を $x=i, y=j$ とする。

$Ps[j-1][i]$ にビットがなければ、 $y=j+1$

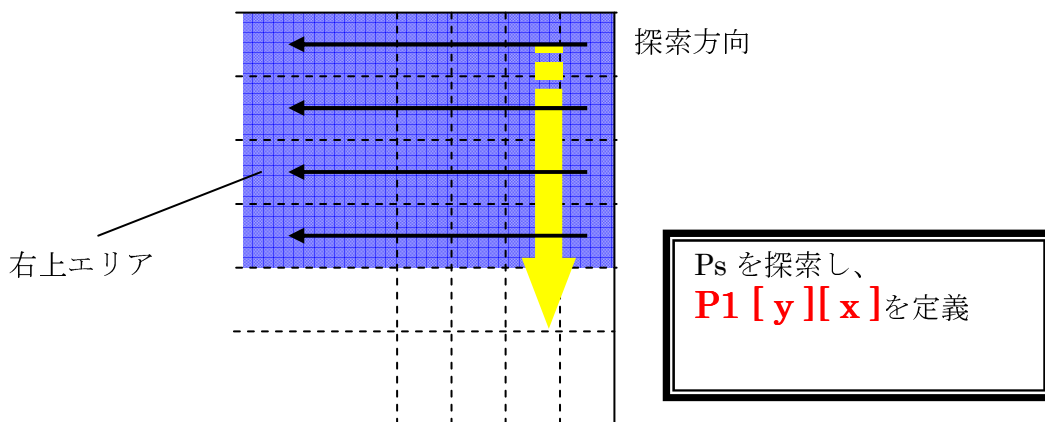
$Ps[j][i-1]$ にビットがなければ、 $x=i+1$

i 列



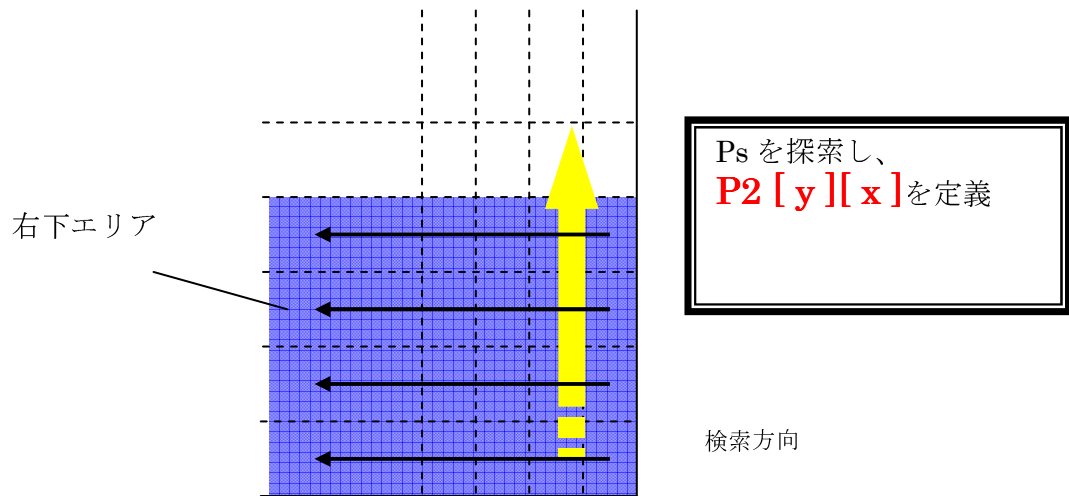
(2) 右上エリア

右上コーナーから、下図のように横方向に探索し、**右上コーナーポイント ($P1$)** を定義する。



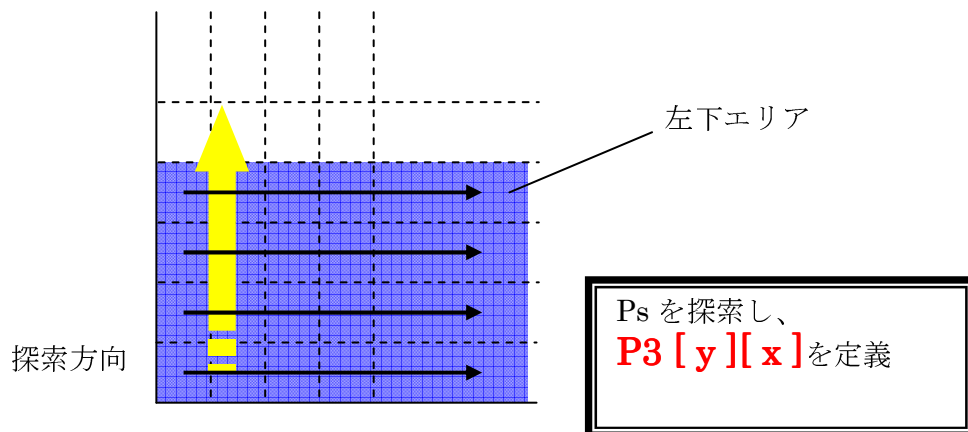
(3) 右下エリア

右下コーナーから、下図のように横方向に探索し、右下コーナーポイント (P2) を定義する。



(4) 左下エリア

左下コーナーから、下図のように横方向に探索し、左下コーナーポイント (P3) を定義する。



【原点の決定】

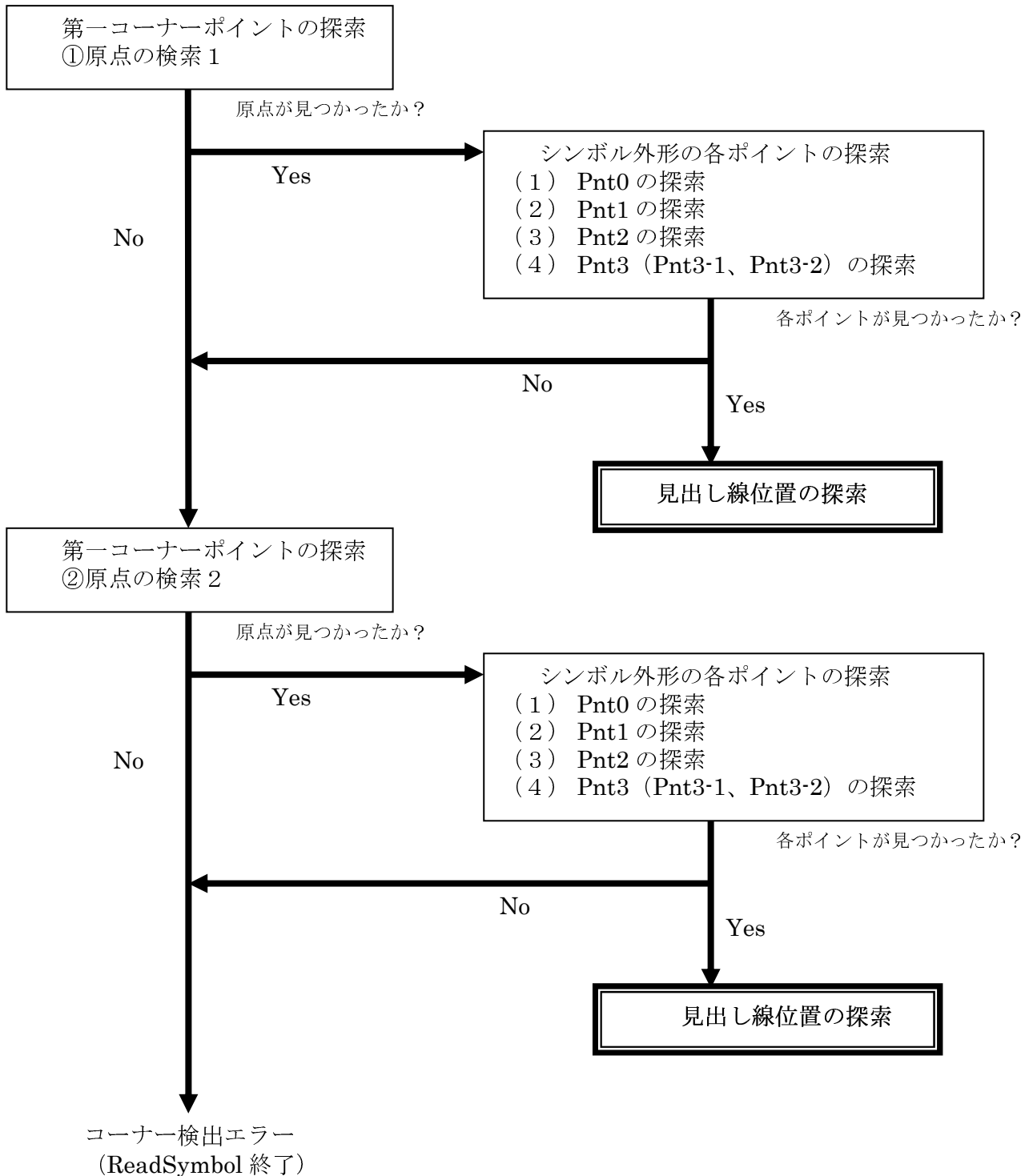
- 4 点の検索1 【原点の決定】と同様の手順で、原点を決定する。

4-3 シンボル外形の各ポイント探索

「第一コーナーポイントの探索」で探索した点 Pt0、Pt1、Pt2 を基に、四隅の点 (Pnt0、Pnt1、Pnt2、Pnt3) を探索する。

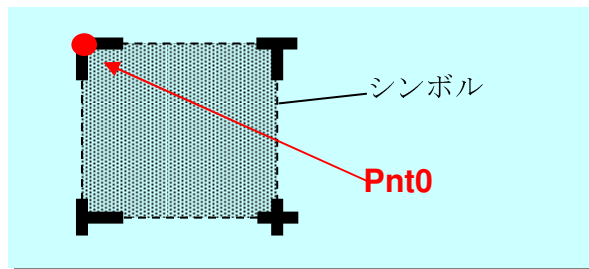
【四隅の点の探索】

シンボル外形の各ポイントは、以下の手順で探索する。



(1) Pnt0

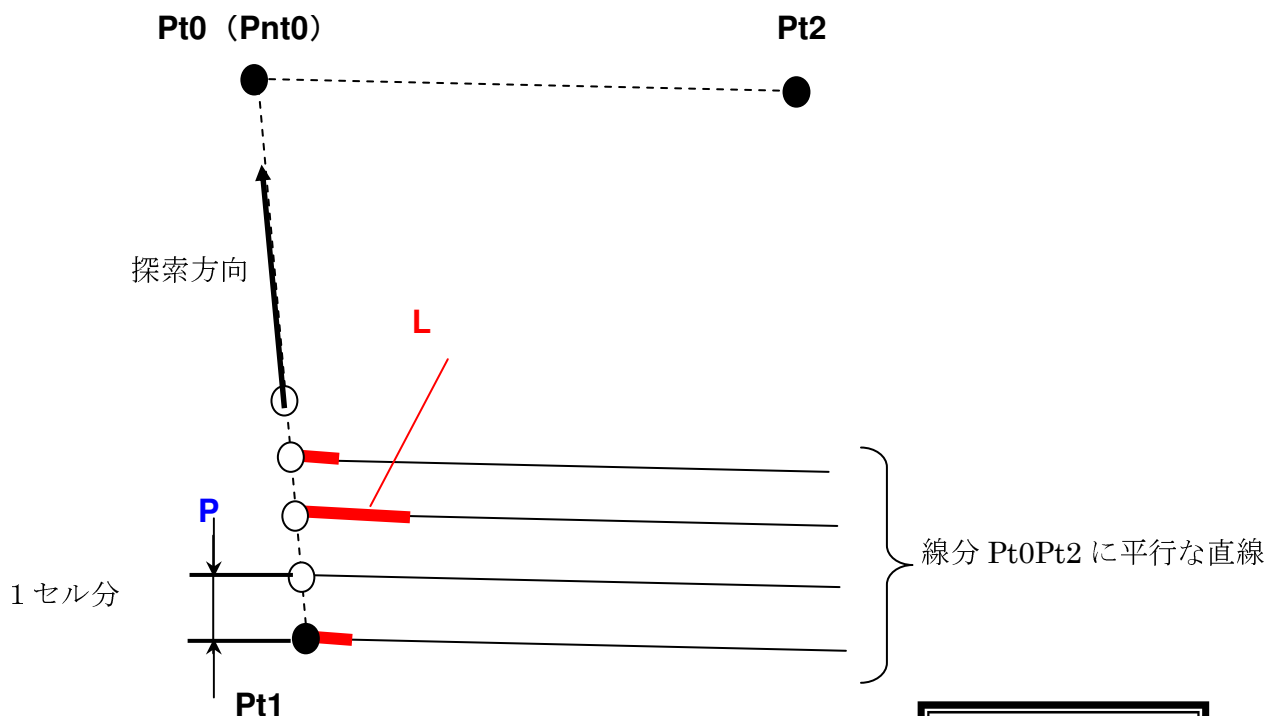
Pnt0 は、原点 Pt0 と同じ点とする。



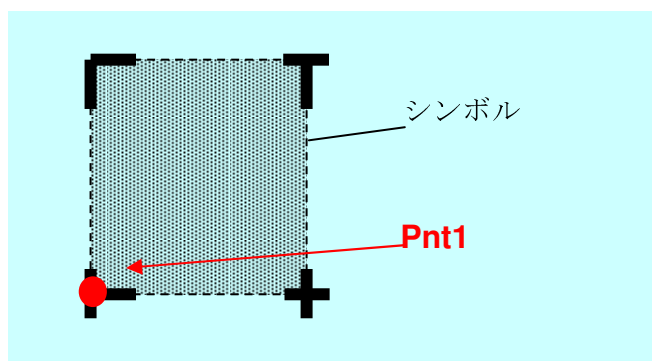
(2) Pnt1

線分 Pt1Pt0 上を Pt1 から Pt0 方向に、1セル間隔で 5セル分探索する。

探索点 P から、線分 Pt0Pt2 に平行な直線を引き、その直線上でビットが連続して存在する長さを L とする。L が 6セル以上であれば、探索点 P を Pnt1 とする。

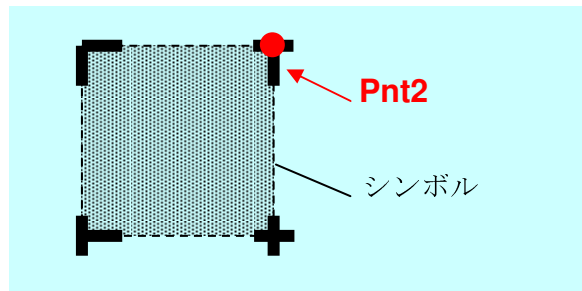
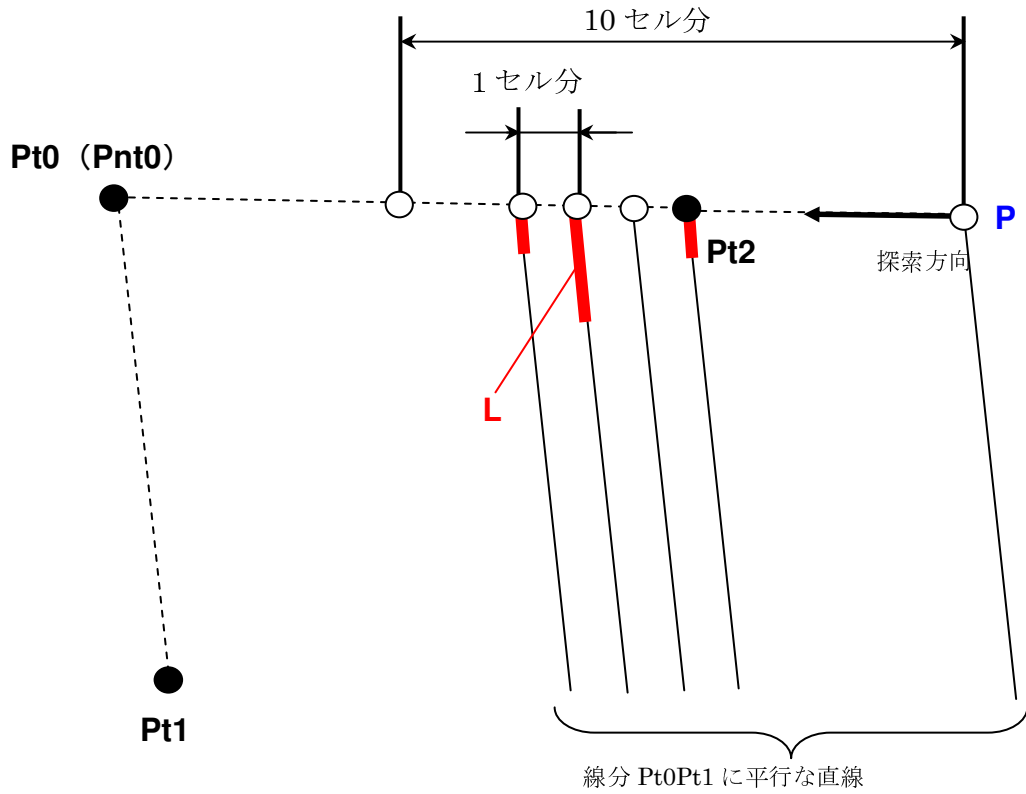


Pnt1 を定義



(3) Pnt2

下図のように点 P から、点 Pt2 の Pt0Pt2 方向±5 セル区間で探索する。
探索点 P から、線分 Pt0Pt1 に平行な直線を引き、その直線状でビットが連続して存在する長さを L とする。L が 6 セル以上であれば、探索点 P を Pnt2 とする。

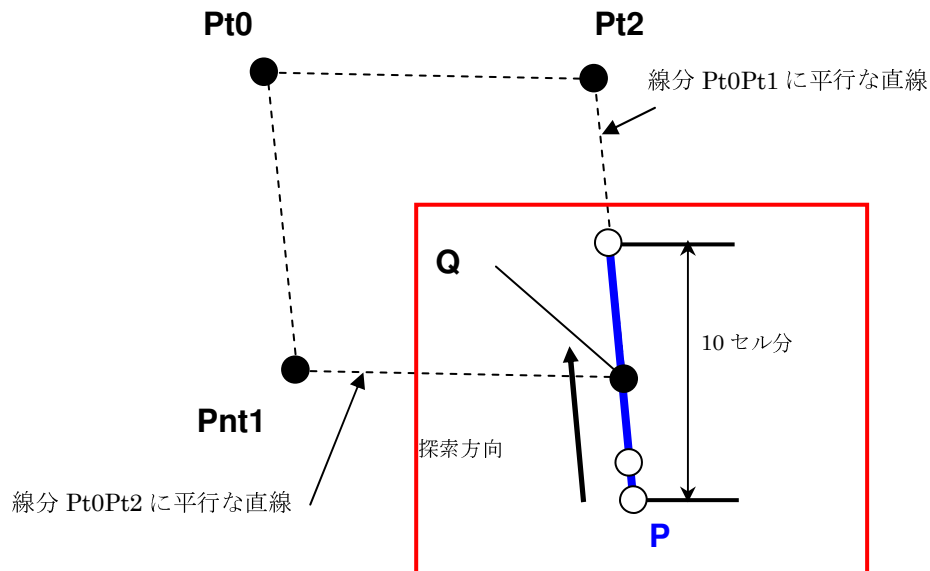


Pnt2 を定義

(4) Pnt3

Pnt1 から線分 Pt0Pt2 に平行な線分の端点 Pnt3-1 を探索する。

下図のように点 P から、点 Q の Pt2Q 方向±5セル区間で探索する。ここで、点 Q は、Pt2 を始点とした Pt0Pnt1 と平行な直線と、Pnt1 を始点とした Pt0Pt2 に平行な直線の交点とする。

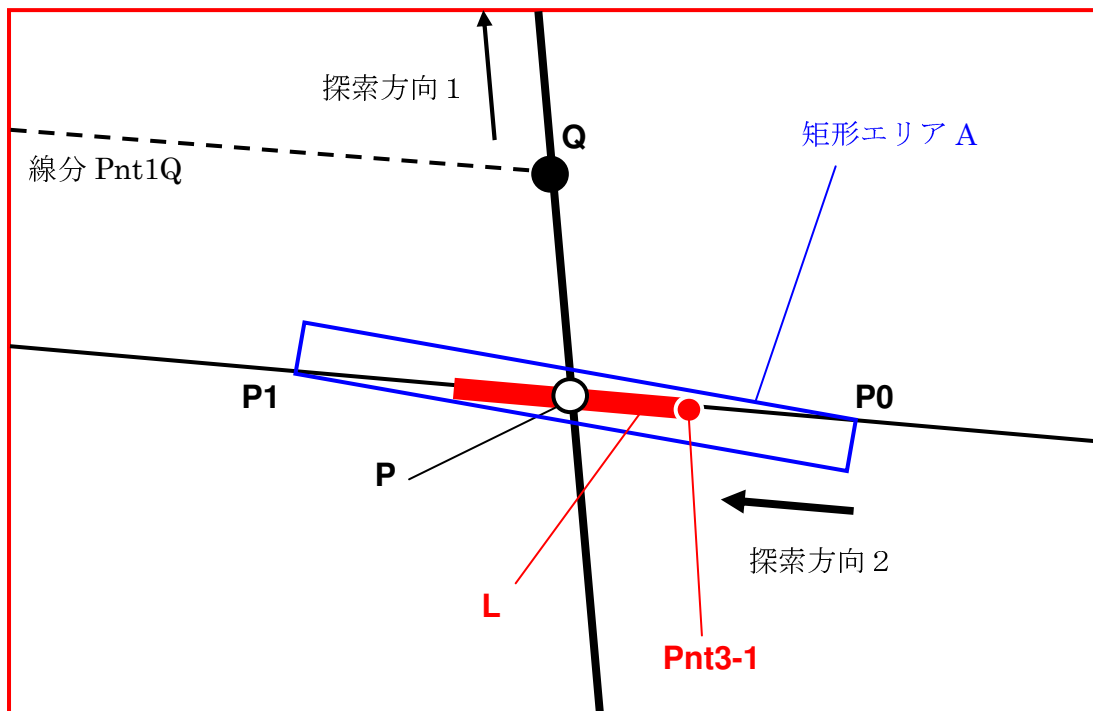


【端点の探索】

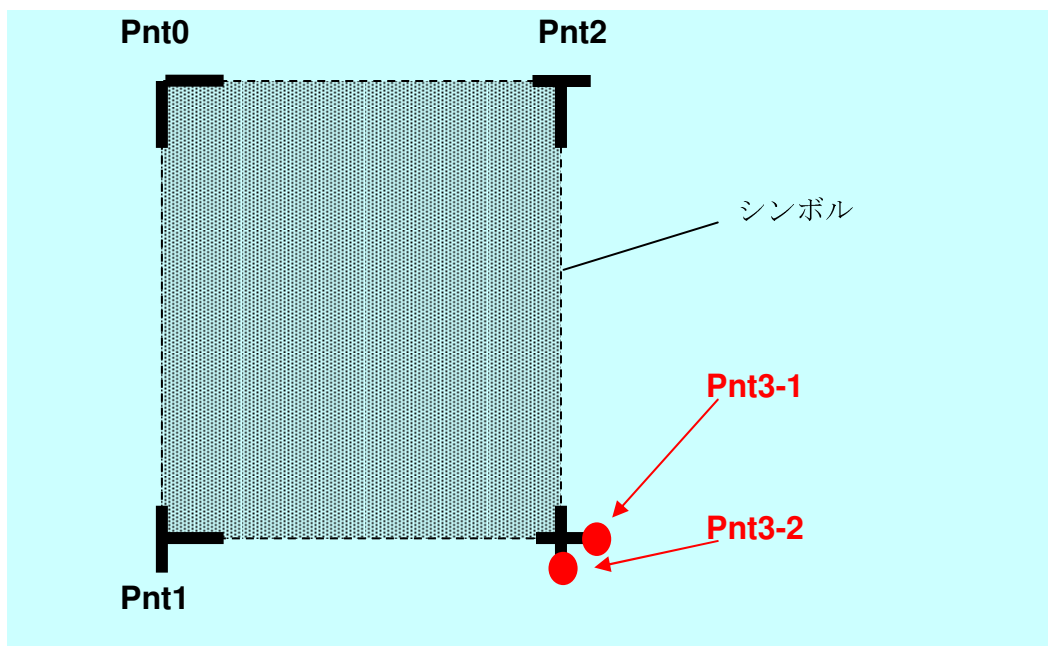
点 P から探索する（探索方向 1）とき、探索点 P を中心に矩形エリア A を設定し、エリア内で端点 Pnt3-1 を探索する。

ここで、矩形エリア A は、点 P を通り線分 Pnt1Q と平行な直線上で、点 P を中心に長さが 8.5 セルとなる点 P0、P1 を対角点とするエリアである。

矩形エリア A の頂点 P0 から線分 P0P1 上を探索し（探索方向 2）、その線分上でビットが連続して存在する最大長区間の長さを L とする。L が 4 セル以上であれば、連続開始点を Pnt3-1 とする。



Pnt2 から線分 Pt0Pt1 に平行な線分の端点 Pnt3-2 を探索する。
 (探索手順は、Pnt3-1 の探索時と同様)



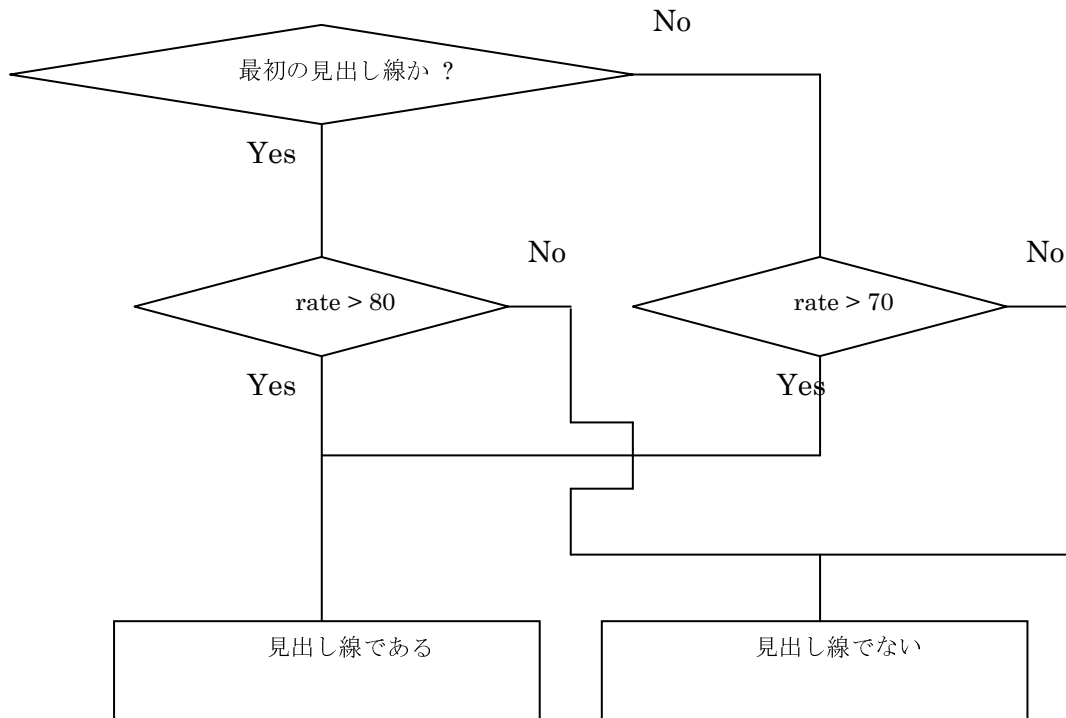
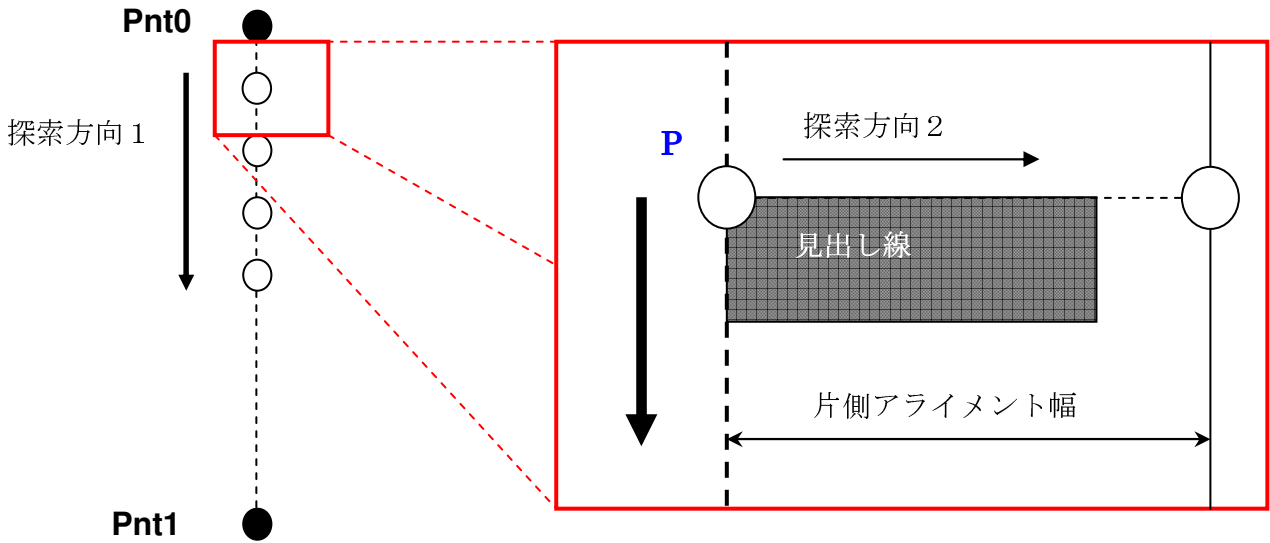
4-4 見出し線位置の探索

「シンボル外形の各ポイント探索」で探索した点 Pnt0、Pnt1、Pnt2、Pnt1-3、Pnt2-3 を基に、ユニットの見出し線位置を探索する。

① 列見出し線位置の探索

線分 Pnt0Pnt1 上を Pnt0 から Pnt1 方向に探索し（探索方向 1）、左手垂直方向に見出し線を探る（探索方向 2）。同様に、線分 Pnt2Pnt2-3 上を Pnt2 から Pnt2-3 方向に探索し、左手垂直方向に見出し線を探る。

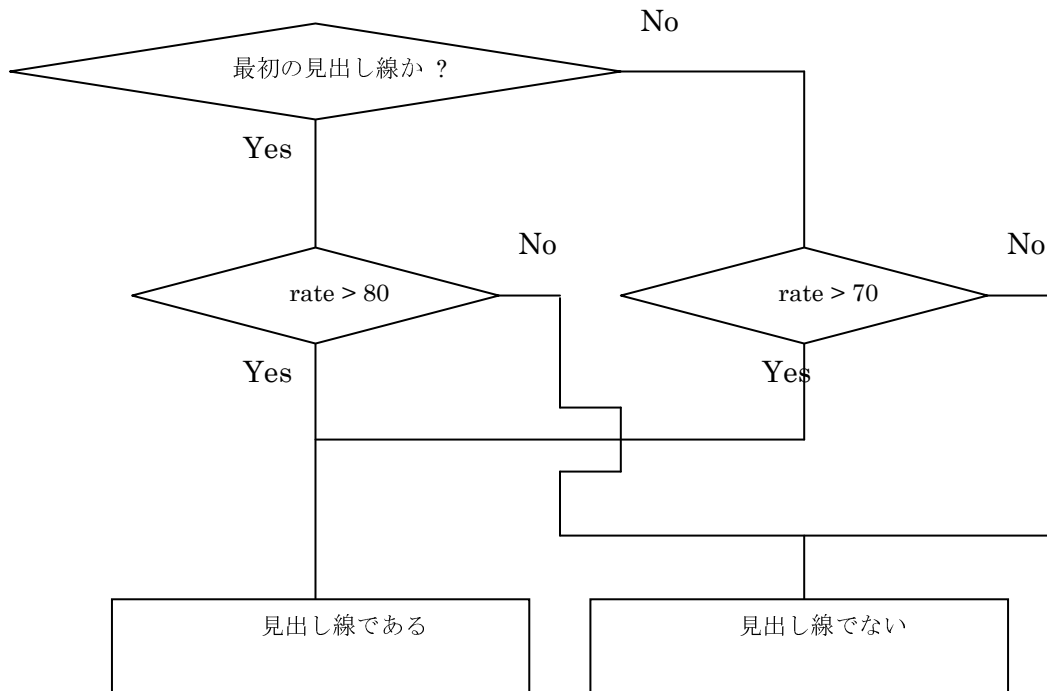
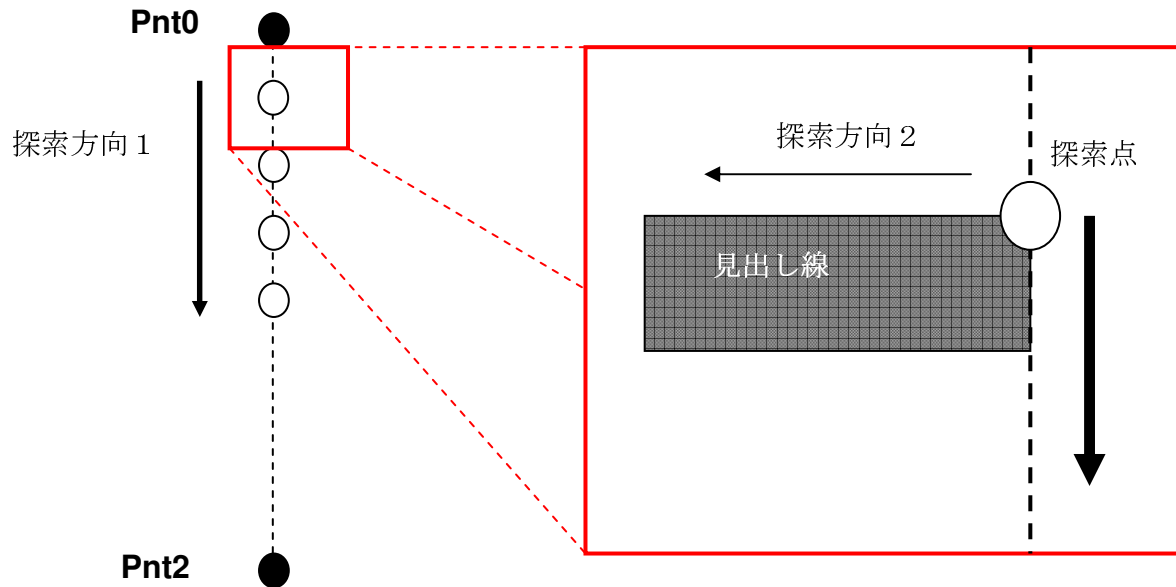
Bresenham の線分発生アルゴリズムを用いて、P、Q を結ぶ線分 PQ 上の点（座標）を生成し、その点のビットの割合 rate(%)を計算する。ビットの割合により見出し線を判定する。



② 行見出し線位置の探索

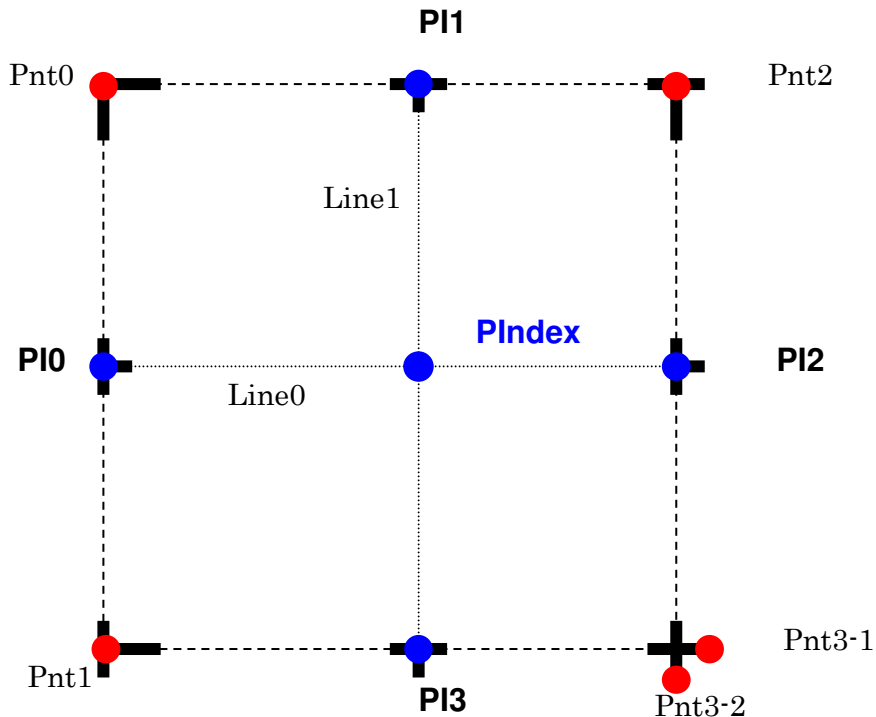
線分 Pnt0Pnt2 上を Pnt0 から Pnt2 方向に探索し（探索方向 1）、右手垂直方向に見出し線を探る探索方向 2）。同様に、線分 Pnt1Pnt1-3 上を Pnt1 から Pnt1-3 方向に探索し、右手垂直方向に見出し線を探る。

見出し線の判定方法は、①列見出し線位置の探索と同様である。



4-5 交差点の計算

「見出し線位置の探索」で探索した見出し線位置 (PI1、PI2、PI3、PI4) から、各見出し線の交点 PIndex を計算する。



【PIndex の計算】

見出し位置 Pnt0、Pnt1、Pnt2、Pnt3 から交点 Pindex(X,Y)を求める。

2点 Pnt0(x0,y0)、Pnt2(x2,y2)を通る直線 Line0 及び 2点 Pnt1(x1,y1)、Pnt3(x3,y3)を通る直線 Line1 を求める。

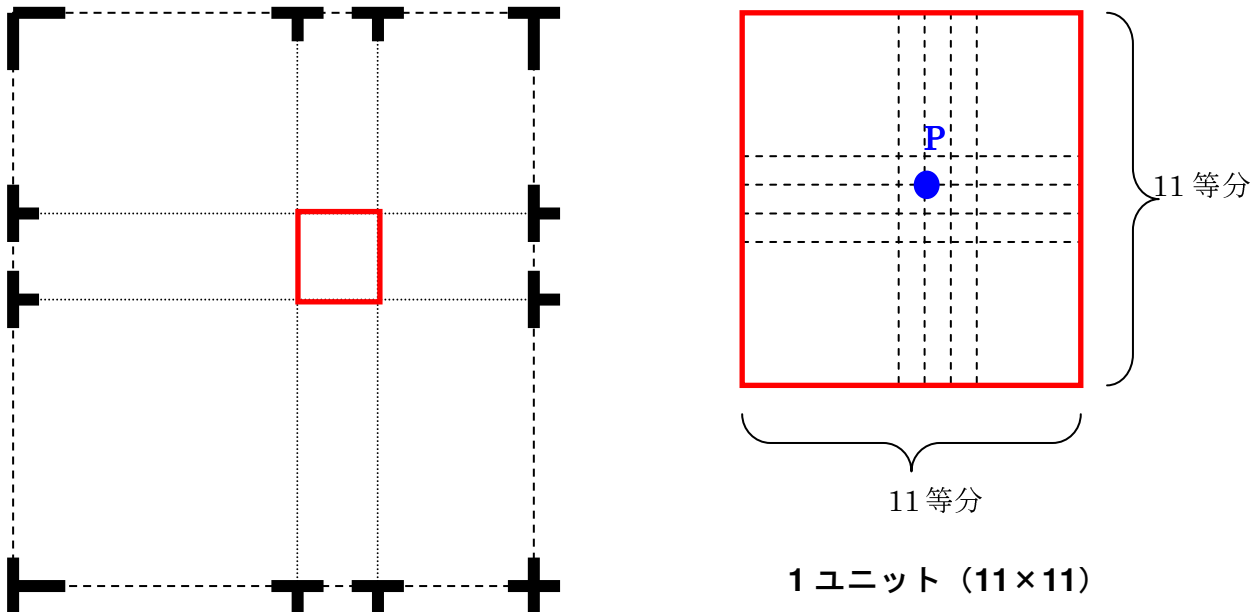
$$\begin{cases} y = a_0 x + b_0 & (\text{Line0}) \\ y = a_1 x + b_1 & (\text{Line1}) \end{cases}$$

Line0 と Line1 の交点 Pindex(X,Y)を求める。

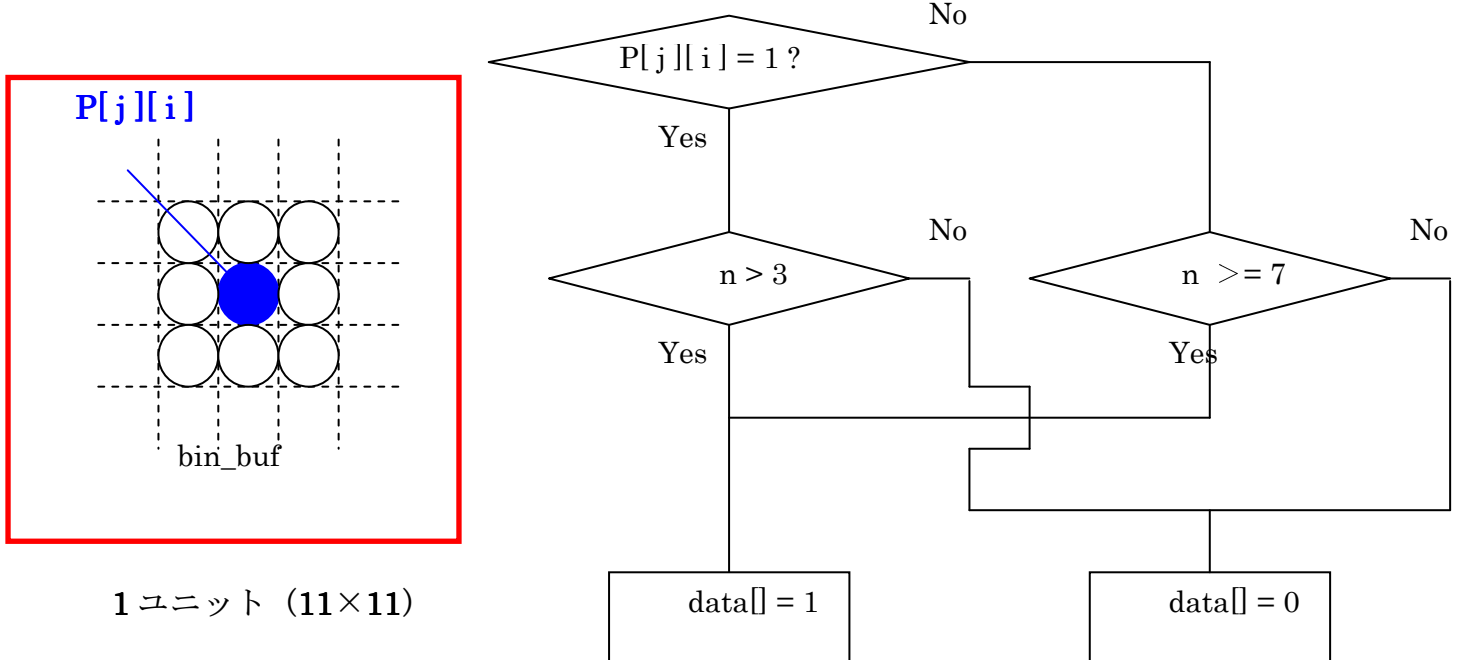
$$\begin{cases} X = (b_1 - b_0) / (a_1 - a_0) \\ Y = a_0 X + b_0 \end{cases}$$

4-6 ビットサンプリング

各見出し線により縦横に囲われたデータブロック（1ユニット）内を、等分に分割して各セルの中心座標 P を求める。



3のピクセルを読み出し、各ピクセルに重みを付け 1 or 0 の判定を行い、ビット列データ $data[]$ を作成する。

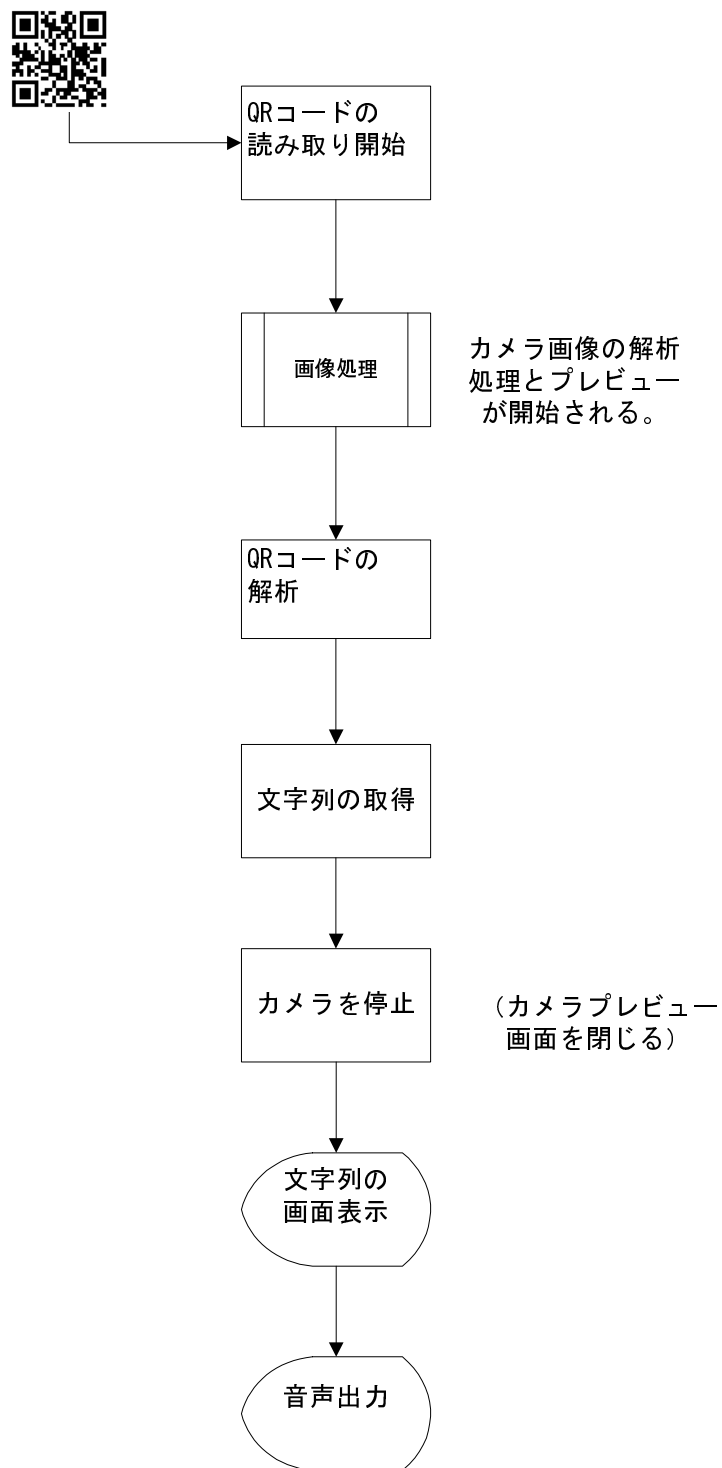


n : 3×3 ピクセル内のビット数

5. QRコードの取り込み

ここでは、新たに追加した QR コードを取り込み、解析し文字列を画面に表示、音声出力までの流れを説明する。

5-1 処理の流れ

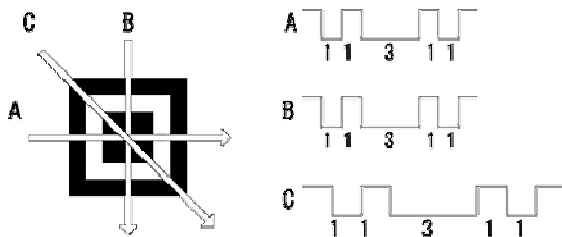


5-2 QRコードの読み取り

QRコードの切り出しシンボルを検出しフォーマット情報を読みデコードする。

QRコードを構成する最小の単位であるセル(白黒の正方形)の組み合わせで現わされているものを読み取り文字化、音声化させる。

○切り出しシンボル



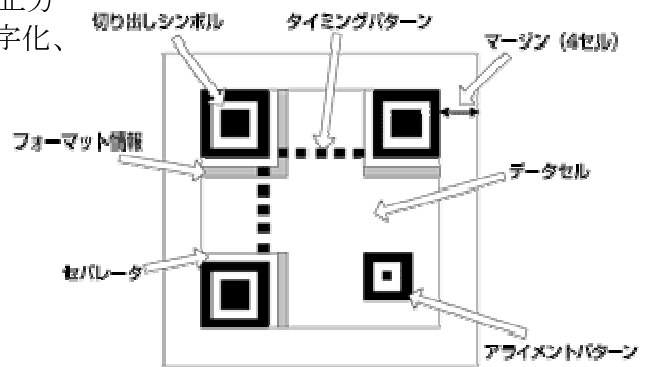
3コーナー (A,B,C) に配置される部分を検索し、位置を正確に認識させる。

○タイミングパターン

シンボル内のモジュール座標を認識し解析する。

○フォーマット情報

まず最初にここを読み誤り訂正率、マスクパターンを認識する。



QRコードの構成

5-3 QRコードの解析

QRコードの切り出しシンボルを検出しフォーマット情報、タイミングパターン、セルを抽出し解析して文字変換したものを画面に表示、音声出力を行う事で、今回のQRコード読み取り機能を追加した。

1 PC 接続型

仕様

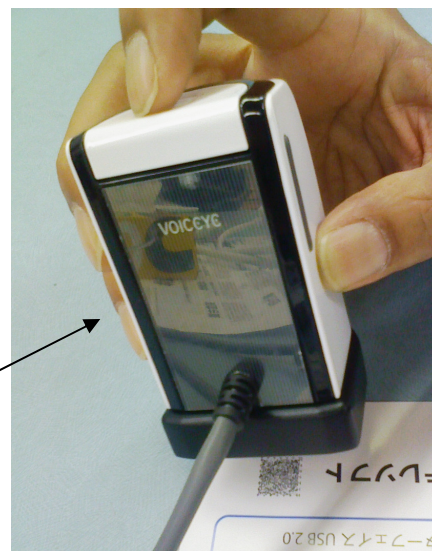
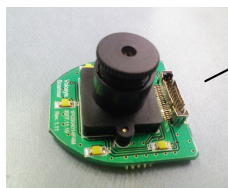
カメラ : CMOS

I/F : USB 2.0(コード 1.5m)

サイズ : 85mm (H)

50mm (W)

52mm (D)



PC アプリケーション

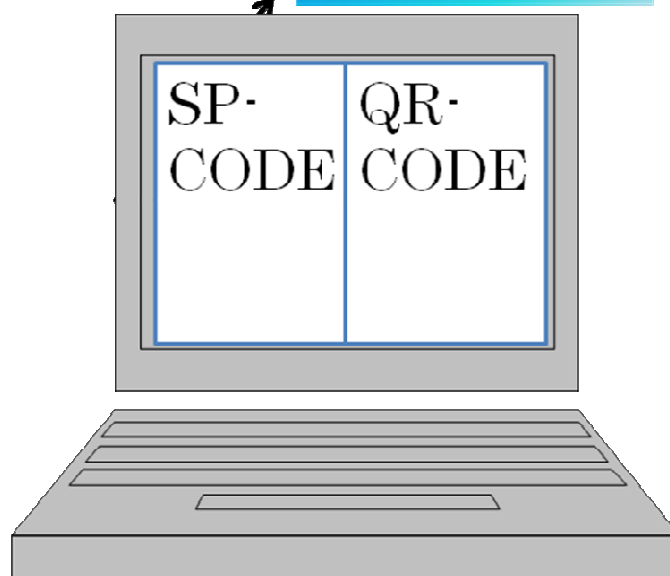
初期メニュー

ここでは、まず音声コード読み上げプログラムをアイコンから起動、或いは、プログラムから、または、スタートアップから Windows 起動時に立ち上がるように設定し、2種類のプログラムを選択するメニュー画面を表示させる。



音声コードReader

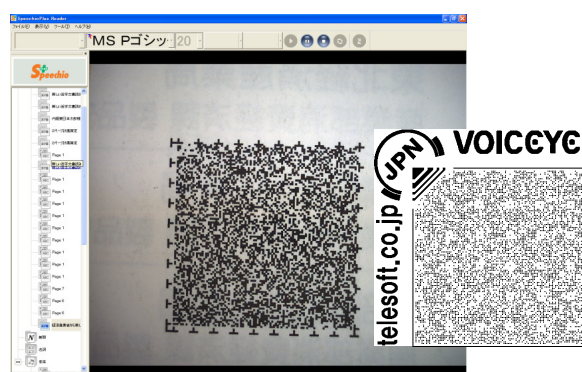
タッチパネルにも対応



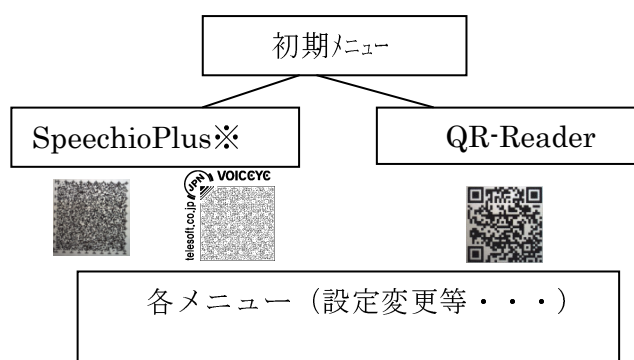
SP-CODE Reader (SpeechioPlus) QR-CODE Reader (新たに開発)

初期メニューで SP-CODE を選択すると
SpeechioPlus が起動する。
QR-CODE を選択すると QR-Reader が
起動する。

SpeechioPlus 起動画面
※SP コードと新ハイスpekコード
の読み取り



QR-CODE Reader 起動画面



※SpeechioPlus のプログラムは新ハイスpekコードを
読ませた場合自動判別する為、同じメニューから読上げが
出来るようになっている。

<例>

ア. SP コードと新ハイスpekコード

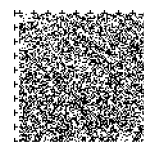
次頁にある、音声コード入りのサンプルを PC 接続型音声装置読み取り機器で読ます事により音声
の出力及び、PC 画面上への出力ができる。

※上部が新ハイスpekコードで下部が SP コード

サンプル

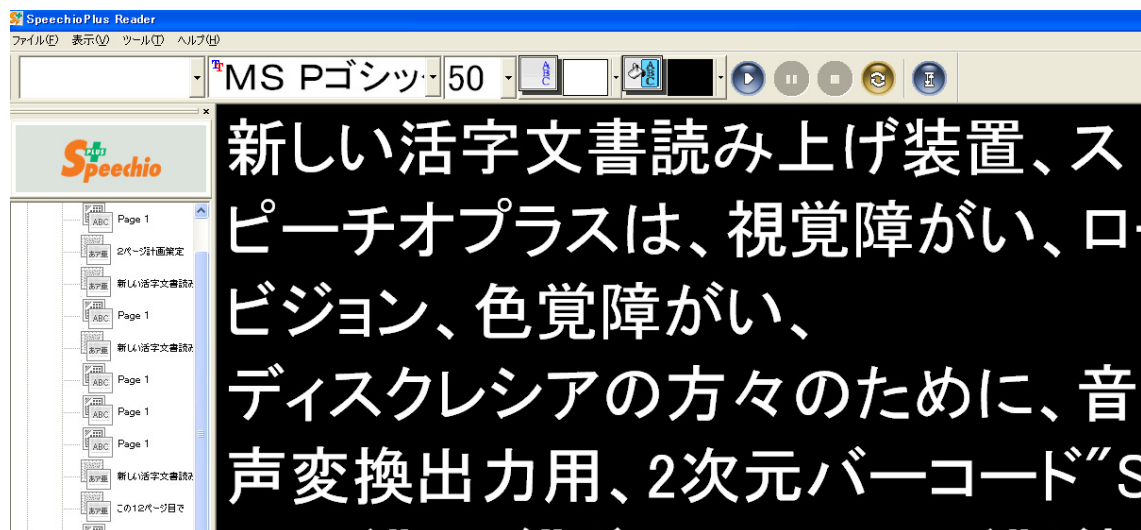


新しい活字文書読み上げ装置、スピーチオプラスは、視覚障がい、ロービジョン、色覚障がい、ディスレシヤの方々のために、音声変換出力用、2次元バーコード”SPコード”及び”新ハイスペックコード”が挿入された印刷物をスキャナーとパソコンを使って読み取り、印刷物の内容をパソコン画面上にテキスト表示するとともに、音声で読み上げる装置です。



➔ ♪ 音声出力 ♪、画面表示（弱視向けに拡大反転表示）

※SP コード、新ハイスペックコードの表示画面は、共通のプログラムなので同じものになる。



また、前頁のサンプルにて同サイズの SP コードと新ハイスペックコードを組み込んだが、この同サイズで SP コードは最大約 800 文字を取り込めるが、新ハイスペックコードの場合は、最大約 1,500 文字の情報を取り込める。

イ. QR コード



<http://www.telesoft.co.jp/>



上記 QR コード内に納められた文字情報が音声出力され、右にある PC 上の画面に表示される。

(弱視向けに拡大反転表示も可能)



1-1 PC 接続型機器：今後の展開

今回、開発した PC 接続型は、3 種類のコード（SP コード、新ハイスペックコード、QR コード）を読み取り、音声出力、画面出力（弱視障害者向け）出来るようになった。今後、視覚障害者にとって利用出来る環境が広がって行くと考えられる。

更に、個人だけではなく、自治体の窓口や、公共の場に設置される事により、視覚障害者が出先でも健常者と同じ様に情報をその場で得られる事になるので、自治体、公共施設へ展開し、この機器の普及を広げて行きたい。

また、QR コードに関し今後は、Web の URL を読み取った場合、直ぐにサイトを開き音声でガイドし、Web からの情報も簡単に得られる様にしていく。それを行う事により視覚障害者にとって効果的な情報機器となると考えられる。

2 携帯型

2-1 仕様

OS	WinCE 6.0
メインメモリー	256MB
フラッシュメモリー	4GB
SD カードスロット	32GB まで対応
マイク	内蔵/マイクロフォンジャック
バーコードスキャナ	5000LED Aimer/Bar CODE(UPC)
SP CODE スキャナ	1.3M Pixel /SP 2D code
USB	クライアント 1 (USB 2.0) HOST 1 (1.0)
バッテリー	1750mA (フル充電 8 時間使用)
充電	AC アダプター/USB 充電
サイズ	115(W)×63(D)×27(H) ※スキャナ部 40(H)
重さ	170g

音声コード以外に、バーコードを読ませる機能を考えており、バーコードの情報を音声化させる。

バーコードを音声コード化する事で、より音声コードが身近になり、広がり拍車をかけると考えられる。

その他の拡張機能

- ・デイジー機能 (デジタル録音図書)
- ・音声メモ
- ・MP3 プレイヤー
- ・色認識 (色を音声で案内)



2-2 携帯型機器：今後の展開

携帯型の商品化

音声コード以外にバーコードを読み取りその商品の名称やその商品に関する内容を音声にする事を検討。

1. 本機器でバーコードをスキャンし商品コードを取得する。
2. 取得した商品コードを機器内のプログラムで機器内にある商品データベース*から商品名や付属情報を拾う
3. 本機器で音声出力し案内を行う。

*商品データベースは JICFS/IFDB (Jan Item Code File Service/Integrated Flexible Data Base) を利用する。この JICFS/IFDB (JAN コード統合商品情報データベース)は財団法人流通システム開発センターが管理、サービスを行っており、JAN コードとこれに付随する商品情報を一元的に管理するデータベースサービスである。

JICFS/IFDB は、小売業における POS システムや EOS などの導入運用に必要な商品マスタや、小売業と卸売業間のオンライン受発注処理で利用される商品情報など、流通情報化において作成負荷が大きい商品マスタ情報を収集整備し、誰もが低コストで迅速に正確な商品情報を得られる事を目的とされたものである。

この JICFS/IFDB を利用しシステム化するには、流通システム開発センターの協力が必要になる。今後は、こことの調整も必要となる。

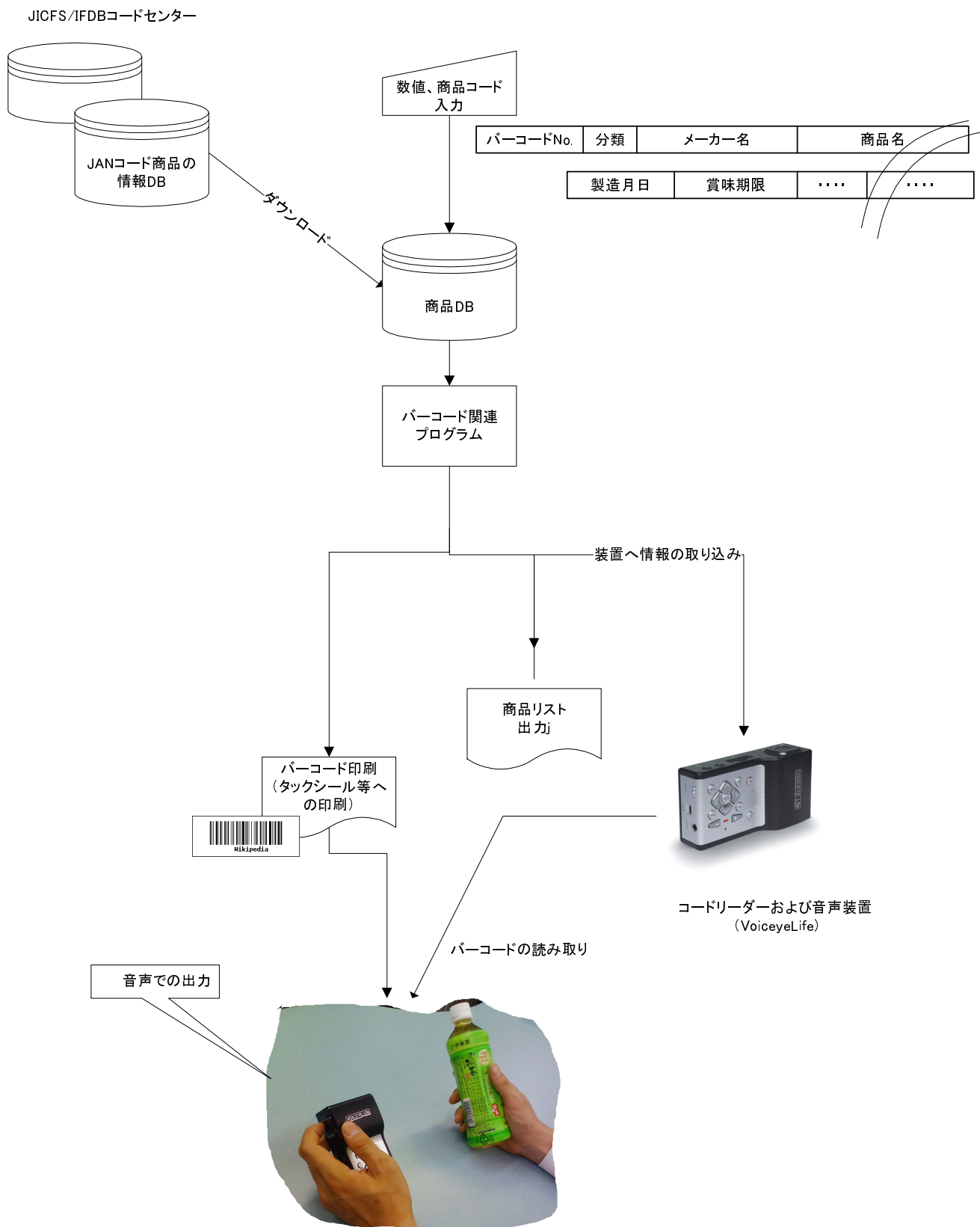
また、DB の組み込みや機器への情報組み込みや、個人個人が生活で利用する為の独自のコード作成・出力ツールの作成が必要となる。

問題点として「バーコードの位置が視覚障害者はわかるか？」という点がある。

当然、位置を認識できない。但し、缶飲料等は側面に縦向きに印刷されているので缶の側面にバーコードリーダーを向け、缶を回転させれば、読み取りが可能となる。

他製品については、今後、バーコード位置の統一化を図る様、働きかける必要があるが、ゆっくりとこの機器を商品に向け動かして行けば、読み取る事ができる。

1次元バーコード（商品コード）生成の流れ



1 SP コードの形状

1-1 各部の名称

SP コードの形状は、以下に示す図のようになっている。また、以下に SP コードを構成する各部の名称を示す。

- ① セル
二次元コードを形成する最小のドットを「セル」と呼ぶ。
- ② ユニット
セルの 11×11 単位を「ユニット」と呼ぶ。
- ③ データマトリクス
SP コード内部のデータ格納領域を「データマトリクス」と呼ぶ。
- ④ 見出し線
SP コードの四方を取り囲む凸型の部分を、「見出し線 (アラインメント)」と呼ぶ。見出し線の太さの幅は 1 セルで構成されている。

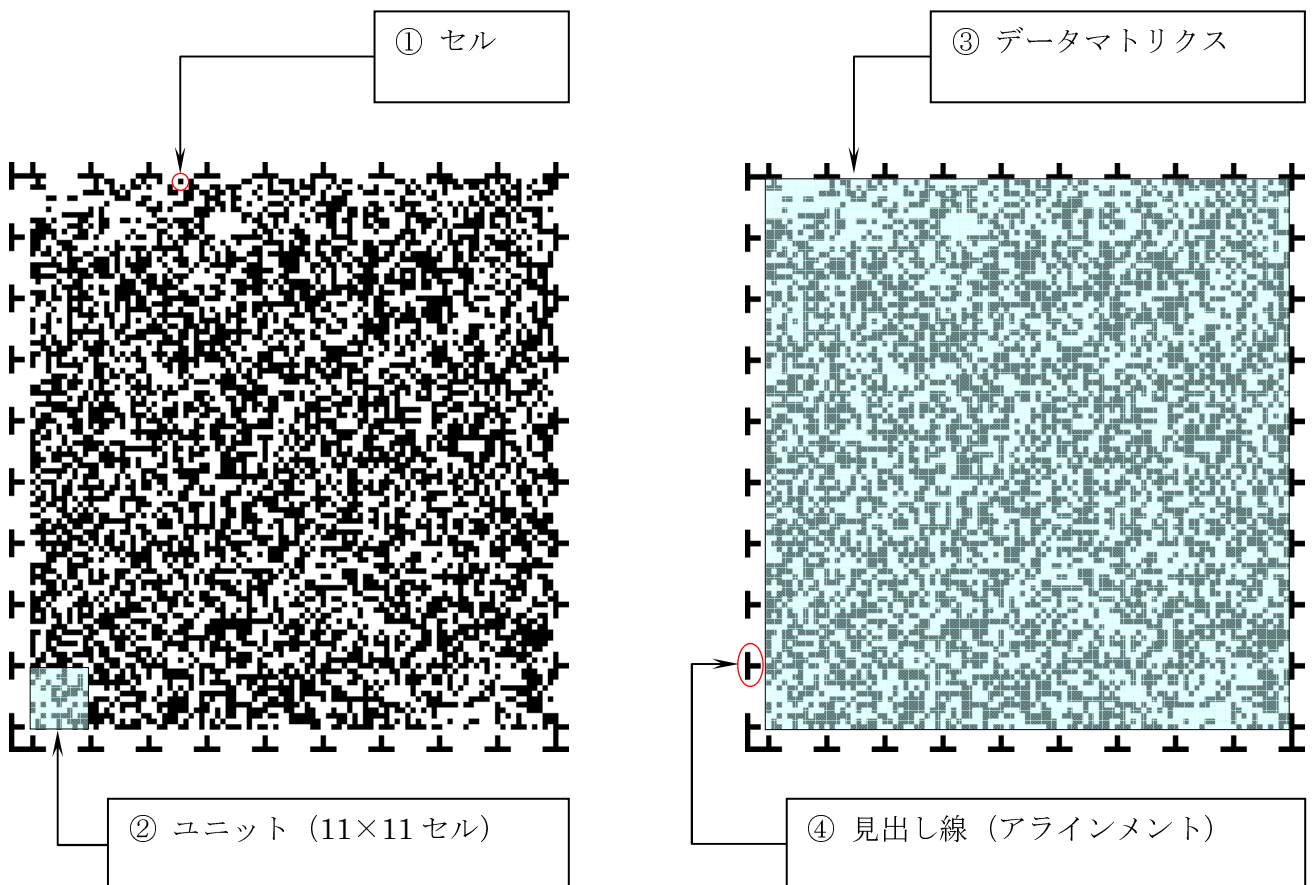


図2 SP コードの各部の名称

左図と右図は同じ M サイズの SP コード (M サイズについては「1-2 サイズとデータ容量」参照のこと)。図は説明のために拡大して表示。

1-2 サイズとデータ容量

SP コードには、XS・S・M・L の 4 段階のサイズがあり、それぞれの大きさとデータ容量は下表のようになっている。

表 1 SP コードのサイズとデータ容量

サイズ	セル数	ユニット数	印刷時の大きさ (mm)	誤り訂正	データ容量 (全角文字数)
XS	40×40	3×3	6.8×6.8	強	41
				中	48
				弱	51
S	73×73	6×6	12.4×12.4	強	250
				中	298
				弱	329
M	106×106	9×9	17.9×17.9	強	651
				中	768
				弱	840
L	117×117	10×10	19.8×19.8	強	793
				中	921
				弱	1027

セル数

見出し線を含めたセル数。

印刷時の大きさ

SpeechioSymbol 関数（「2-3 シンボル作成関数：SpeechioSymbol」参照）にて作成された BMP (Bitmap) 画像を、600dpi にて印刷出力したときの大きさ。

誤り訂正

誤り訂正には、強・中・弱の 3 種類があり、SP コード作成時の SpeechioSymbol 関数で指定できる（「2-2 エンコード関数：SpeechioEncode」参照）。

データ容量

目安としての数字であり、テキストデータの内容によって圧縮率が異なるため、文字数は一定ではない。

2 ライブラリ (Speechio.dll)

SP コードは、専用のライブラリ「Speechio.dll」を使用して、Microsoft Windows 環境 (Windows XP 日本語版 以降) で容易に作成することができる。本章では、API 関数の仕様説明のみを行い、dll 内の関数の呼び出し方法など、Windows プログラミングに関する説明は、使用例、およびサンプルプログラムにとどめるものとする。

SP コードの作成は、SpeechioEncode 関数によるテキストデータのエンコードと、SpeechioSymbol 関数によるエンコードデータのシンボル化の順で行う。SpeechioEncode 関数では入力テキストデータがエンコードされ、データが bit_string へ格納される。ここで、bit_string は"0"と"1"からなる文字配列であり、SP コードの左上から右下へのセルをあらわす中間データである。SpeechioSymbol 関数では、この bit_string と SP コードの縦横のセル数から、SP コードの画像データが作成され、作成された画像データはクリップボードへ転送される。

2-1 入力仕様

入力文字コードは、シフト JIS コードとし、制御コード (0x01~0x1f, 0x7f) は基本的に含めないものとする (「4 入力テキストの整形処理」参照)。シフト JIS コードには、半角 ASCII 文字 (7bit Ascii, 0x20~0x7e)、半角カナ文字 (JIS X 0201 カタカナ, 0xa1~0xdf) を含む。

2-2 エンコード関数 : SpeechioEncode

```
short __stdcall SpeechioEncode(  
    char data_type[],  
    char cell_type[],  
    char recover_level[],  
    char copyright[],  
    short data_size,  
    unsigned char data_code[],  
    char path_name[],  
    unsigned char bit_string[]  
);
```

機能

入力テキストデータをエンコードする。

引数

data_type	変換対象データの種類 ="T" (日本語テキスト)
cell_type	作成する SP コードのサイズ ="s" (XS) /="S" (S) /="m" (M) /="M" (L)
recover_level	誤り訂正の強さ ="S" (強) /="N" (中) /="P" (弱)
copyright	未使用
data_size	変換対象データのバイト数
data_code	変換対象データ
path_name	未使用
bit_string	エンコード後のデータを返すバッファ

戻り値

>0 正常終了 : bit_string へエンコードデータが格納される
 =0 エラー : エンコードに失敗
 <0 データ容量オーバー : = (オーバーしたバイト数) × -1

補足

- ・エンコードデータが格納される bit_string は、呼び出し側であらかじめメモリ領域を確保しておく必要がある。
- ・bit_string は、SP コードの左上から右下方向へのセルを表す中間データで、"0" (0x30) と "1" (0x31) からなる配列として生成される。"0"が SP コードの白のセルを、"1"が黒のセルを表す。

2-3 シンボル作成関数 : SpeechioSymbol

```
short __stdcall SpeechioSymbol(
    short col,
    short row,
    unsigned char symbol_data[]
);
```

機能

エンコードデータからの SP コード画像を作成 (シンボル化) する。

引数

col シンボルの横方向のセル数
 =40 (XS) /=73 (S) /=106 (M) /=117 (L)
 row シンボルの縦方向のセル数
 =40 (XS) /=73 (S) /=106 (M) /=117 (L)
 symbol_data SpeechioEncode 関数で作成した bit_string を指定

戻り値

=0 正常終了
 <0 エラー

補足

- ・成功時は SP コードの画像データが、BMP 形式でクリップボードへ転送される。
- ・作成される画像は、SP コードの 1 セルが 4×4 ドットで構成される。

2-4 誤り訂正

誤り訂正は、コードの読み取りに問題がなければ、標準の「中」を推奨する。コード印刷面が汚れる環境などで使用される場合は、「強」とする必要がある。ただし、この場合は搭載できるデータ容量が小さくなるので注意する必要がある（1-2 サイズとデータ容量：表 1 参照）。

2-5 関数の使用例

エンコード関数・シンボル作成関数について、簡単な使用例を以下に示す。

```
/*
 * エンコード関数・シンボル作成関数の使用例
 * "Speechio.lib"ファイルへリンクする。"Speechio.dll"はパスの通った
 * フォルダ（システムフォルダかプログラムのあるフォルダ）へ配置する。
 */
#include <stdio.h>
#include <string.h>
#include "Speechio.h" /* 関数プロトタイプ宣言、SP_MAX_BIT_SIZE を定義したヘッダ */

int main(void) {
    unsigned char text[] = "SP コードの作成例です。"; /* 入力テキストデータ */
    unsigned char bit_str[SP_MAX_BIT_SIZE]; /* bit_string バッファ */
    short rc; /* 関数の戻り値用変数 */

    /* エンコード処理 */
    rc = SpeechioEncode("T", "m", "N", "", (short)strlen(text), text, "", bit_str);
    if(rc <= 0) {
        if(rc == 0)
            printf("エンコード処理に失敗しました\n");
        else if(rc < 0)
            printf("%hd バイトオーバーしました\n", -rc);
        return 1;
    }

    /* SP コード画像作成処理 */
    rc = SpeechioSymbol(106, 106, bit_str);
    if(rc < 0) {
        printf("画像作成処理に失敗しました\n");
        return 1;
    }

    printf("SP コード画像データがクリップボードに転送されました\n");

    return 0;
}
```

3 SP コード画像の印刷

音声コード読取機器では、台座上に固定された CCD カメラで画像を取り込むため、PC 上で作成した SP コードは、紙面上で適切な大きさと位置となるよう印刷する必要がある。

3-1 画像の大きさ

「SpeechioSymbol 関数」にて作成される BMP 画像は、SP コードの 1 セルが 4×4 ドットで構成されている。音声コード読取機器は、この画像を 600dpi の解像度で印刷した紙面上の SP コードを、精度良く読み取れるよう設計されている（600dpi 以上のレーザープリンタで出力したものと想定）。この場合、紙面上に印刷される SP コードの大きさは、下表のように算出される。

表 2 SP コードの印刷時の大きさ

サイズ	セル数	600dpi 換算での大きさ (mm)
XS	40×40	$40 \times 4 \times 25.4 \div 600 = 6.77$
S	73×73	$73 \times 4 \times 25.4 \div 600 = 12.36$
M	106×106	$106 \times 4 \times 25.4 \div 600 = 17.95$
L	117×117	$117 \times 4 \times 25.4 \div 600 = 19.81$

「SpeechioSymbol 関数」では画像データをクリップボードへ転送するが、これを避けたい場合には、エンコードデータ (bit_string) に対して「SpeechioSymbol 関数」と同等の処理を独自で行う必要がある。解像度を大きくした場合などでも、印刷出力時の SP コードの大きさを、表 2 の値と等しくしないと、音声コード読取機器での読み取り精度に問題が生じる可能性があるため注意が必要である。

3-2 画像の位置と余白

音声コード読取機器では、SP コードの印刷された紙の隅を差し込んで読み取らせるため、適切な位置にコードが印刷されるよう SP コード画像を配置する必要がある。

基本的に SP コードは印刷物の右下に配置するものとし、コードのサイズによらず、コードの中心位置が印刷物の右端から 25mm、下端から 25mm となるように配置する。また、他の文字や図形が SP コード画像と重ならないよう注意が必要である。音声コード読取機器で SP コード画像を識別するためには、コードの周囲に最低 4mm の余白が必要である（図 3 参照）。

印刷物のレイアウトや形状から、右下に配置できない場合は、SP コードを印刷物の左下や右上に配置しても問題はない。この場合、音声コード読取機器では SP コード画像の方向を自動的に識別するため、回転に関しては考慮する必要はないが、印刷物の隅を差し込んで音声コード読取機器で読み取り可能な位置に、コードを配置するように注意が必要である（図 4 参照）。

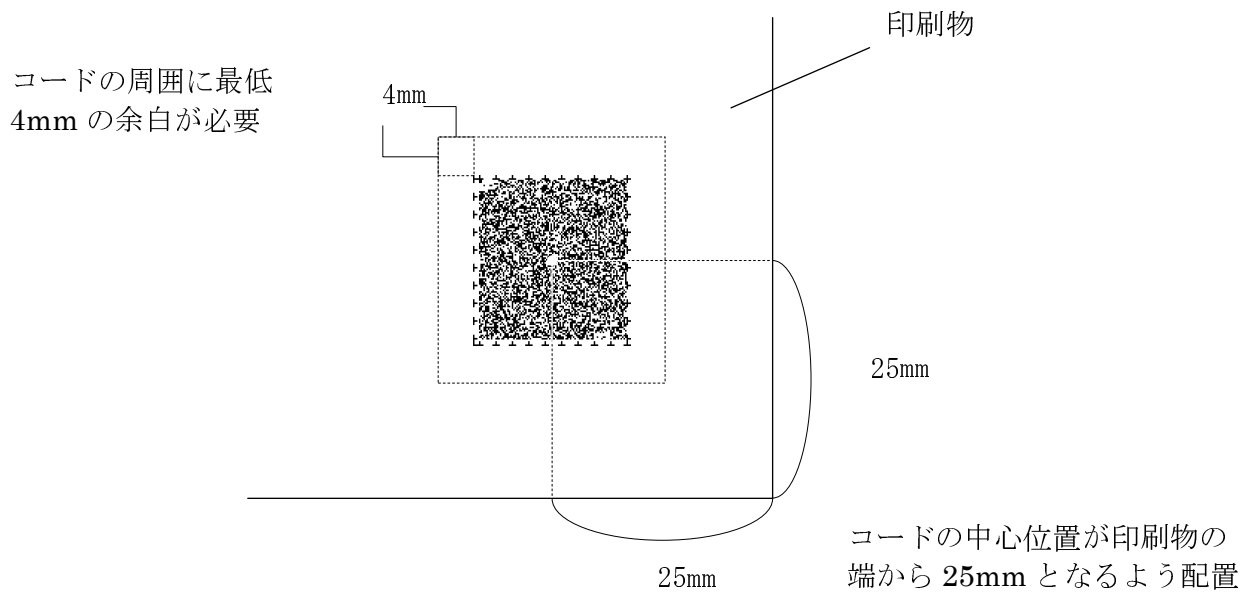


図3 SPコードの位置と余白（右下に配置）

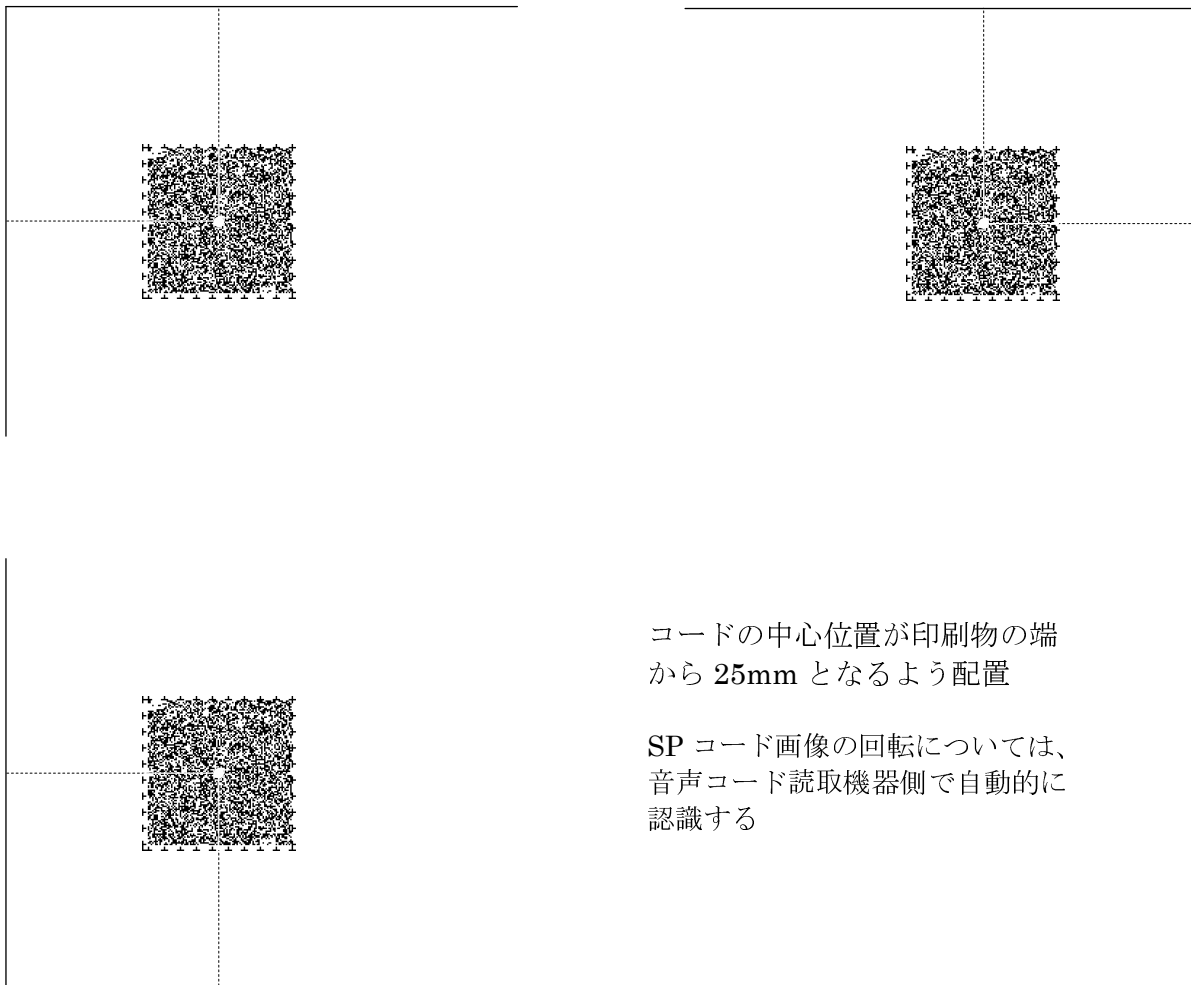


図4 SPコードの配置例

4 入力テキストの整形処理

音声コード読取機器では、SP コード内のテキストを音声合成するため、その中に制御コードなどが含まれていると音声合成プログラム内で不具合が発生する可能性がある。このため、エンコード前にテキストデータ内の制御コードを削除するなどの整形処理が必要となる。

また、音声コード読取機器では、特定の制御コードによる男女の声質の切り替えや、漢字の読み方を指定することなどが可能であり、入力テキストに対して追加情報を付与することができる。

4-1 制御コード等の削除

音声コード読取機器内部での音声合成処理の不具合を避けるため、エンコード前に入力テキストデータ内の制御コードを削除しておく必要がある。

ここでいう制御コードとは、シフト JIS コード内で 0x01~0x1f、0x7f (16 進数表記) の文字コードのことである。ただし、タブ (0x09) ・改行 (0x0a) ・復帰 (0x0d) は除く。これは、音声コード読取機器内部にて、タブを読点に変換し、改行を一度 0x0d0a に統一してから句点に変換しているためである。

また、記号などが音声コード読取機器で意図した読み方とならない場合は、あらかじめ削除しておくことが望ましい。

4-2 読みの指定

音声コード読取機器においてテキストデータを音声出力した際に、目的の読み方で音声出力されない単語がある場合は、その単語の読み方を指定することができる。変換対象のテキストデータに以下の形式で読みを入力する。

(単語:ヨミ)

単語 読み方を指定したい単語 (全角)
ヨミ 単語の読み方を半角カタカナで入力
括弧とコロンは、半角の()と:を使用する。

例 株式会社(廣濟堂:コウサイドウ)は

「カブシキガイシャコウサイドウカ」と音声出力される。

単語の後に半角のコロンを、続けて読みを半角カタカナで入力し、単語の始めから読みの終わりまでを半角の丸カッコで囲む。ただし、単語は全角でなくてはならない。

半角の英数字については、上記の方法で読み方を指定することはできない。この場合、英単語の発音をカタカナで、数字を漢数字で入力するなどの工夫が必要となる。

4-3 音声コード読取機器での声質の切り替え

音声コード読取機器では、テキストデータの先頭に特定の制御コードを付加することで、音声出力に関する制御を行うことが可能である。また、音声コード読取機器では文単位にテキストデータを分割して音声出力を行うため、文単位に音声の制御を行うことができる。例えば、一文目を男性の声で、二文目を女性の声で音声出力することなどができる。音声出力に関する設定の種類と制御コードは、下表の通りである。

表 3 音声出力用の制御コード

音質	制御コード	種類・レベル	備考
声質	^V0	男声 (デフォルト)	^V=0x16
	^V1	女声	
高さ	^H0	レベル 0 (最低)	^H=0x08
	^H1	レベル 1	
	^H2	レベル 2	
	^H3	レベル 3 (男性のデフォルト)	
	^H4	レベル 4 (女性のデフォルト)	
	^H5	レベル 5	
	^H6	レベル 6	
	^H7	レベル 7 (最高)	
大きさ	^P0	レベル 0 (最小)	^P=0x10
	^P1	レベル 1	
	^P2	レベル 2	
	^P3	レベル 3	
	^P4	レベル 4 (デフォルト)	
	^P5	レベル 5	
	^P6	レベル 6	
	^P7	レベル 7 (最大)	

※ 「制御コード+数字」の形式で構成されている。

文頭に制御コードが付加されていない場合は、前文と同じ設定内容で音声出力される。このため、文毎に男女の声を変更する場合は、以下のように制御コードを付加すればよい。

^V0^H3^P4 この文は男性の声で音声出力されます。

^V1^H4 この文は女性の声で音声出力されます。

^V0^H3 この文は再び男性の声で音声出力されます。

制御コードを何も指定しない場合は、表 3 内のデフォルトで設定された声質で音声出力される。つまり、^V0^H3^P4 を指定した場合と同様で、男声・高さレベル 3・大きさレベル 4 の声質となる。

また、制御コードは必ず文の先頭に付加しなければならない。ここで、文とは、音声コード読取機器で判定されるものであり、音声コード読取機器では表 4 の文字を文末として判定する。このため、入力テキストデータの文末を判定して、文単位に分割し、前文と異なる制御コードのみを付加するようにする。

表 4 音声コード読取機器で文末として判定される文字

。)	。))	?)	?)	!))	!))	。
。]	。)]	?]	?]	!)]	!)]	。
。}]	。) }]	? }]	? }]	!) }]	!) }]	?
。}	。) }	? }	? }	!) }	!) }	?
。>	。) >	? >	? >	!) >	!) >	!
。》	。) 》	? 》	? 》	!) 》	!) 》	!
。↓	。) ↓	? ↓	? ↓	!) ↓	!) ↓	CR+LF
。』	。) 』	? 』	? 』	!) 』	!) 』	
。】	。) 】	? 】	? 】	!) 】	!) 】	
。)	。))	?)	?)	!))	!))	
。]	。)]	?]	?]	!)]	!)]	
。}	。) }	? }	? }	!) }	!) }	
。↓	。) ↓	? ↓	? ↓	!) ↓	!) ↓	
。”	。) ”	? ”	? ”	!) ”	!) ”	
。’	。) ’	? ’	? ’	!) ’	!) ’	

※1 連続する文末文字は、一つの文末文字として認識される。

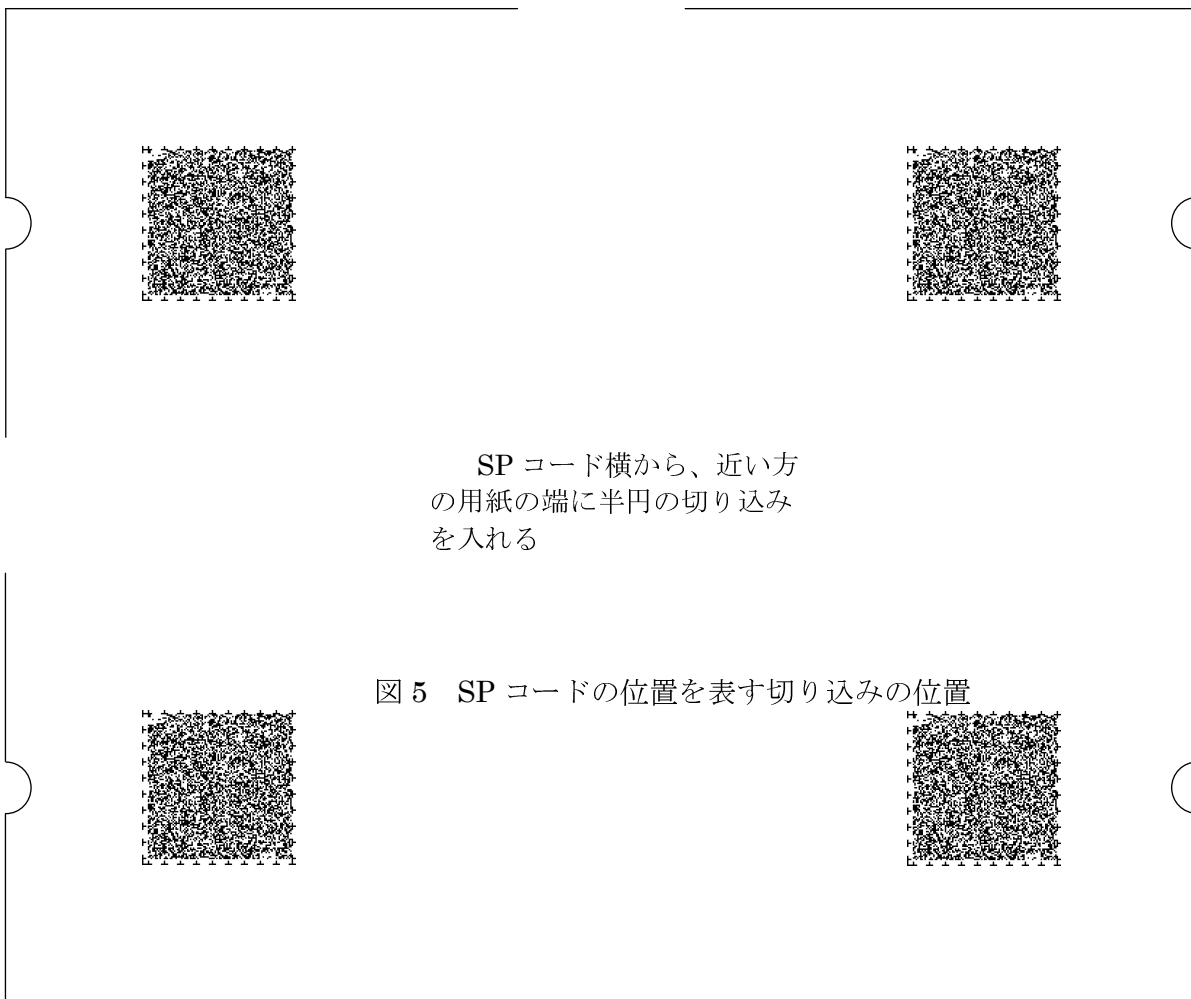
※2 CR=0x0d、LF=0x0a をあらかわす。

5 その他条件

コード生成時の用紙の条件や用紙の加工（切り込み）、印刷条件等を説明する。

5-1 コードの位置を表す切り込みについて

印刷した SP コードの横には、視覚障害者が触覚によりコードの位置を把握できるよう、印刷物に半円の切り込みを入れることを推奨している（下図参照）。切り込みは、市販の穴あけパンチを利用するなどして、印刷物を加工する程度で構わない。



5-2 用紙の指定について

SP コードを印刷する用紙は、一般的なコピー用紙や再生紙等での印刷を推奨する。音声コード読取機器では CCD カメラによって、SP コードを撮影・解析するため、光沢紙のように光を反射する用紙、地の色が濃い色上質紙、凹凸のある加工紙では、場合によっては読み取りに支障をきたす可能性がある。

5-3 プリンタについて

SP コードの印刷は、600dpi 以上のレーザープリンタで出力する必要がある。インクジェットプリンタでは、機種により読み取り可能なコードを印刷できる場合もあるが、インクの滲みによって SP コードが潰れてしまう可能性が高いため、読み取り率は極めて低いというのが現状である。

5-4 SP コードのコピーについて

コピーされた SP コードは、基本的に音声コード読取機器での読み取り率は極めて低くなる。コピー機の性能や、コピー元の印刷状態などにより、読み取り可能な場合もあるが、SP コードのコピーは推奨しない。

SP コード プリンタとセルサイズ

プリンタ	ヘッド密度	1セル	1ユニット
インクジェット	600dpi 24dot/1mm	0.17mm	1.87mm
	360dpi 14dot/1mm	0.28mm	3.08mm
レーザープリンタ	600dpi 24dot/1mm	0.17mm	1.87mm
	360dpi 14dot/1mm	0.28mm	3.08mm
熱転写/サーマル	300dpi 12dot/1mm	0.33mm	3.3mm
	200dpi		
	8dot/1mm	0.5mm	5.5mm

※Microsoft、Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

付録2 音声出力用ユーザー辞書登録内容

No.	見出し	読み	品詞	アクセント
1	点筆	てんぴつ	固有名詞	4
2	点毎	てんまい	固有名詞	0
3	白杖	はくじょう	固有名詞	3
4	拗濁音	ようだくおん	固有名詞	2
5	PC-Talk er	ぴーしーとーかー	固有名詞	5
6	ALTAIR	あるていあ	固有名詞	3
7	mma il	えむえむめーる	固有名詞	5
8	95Reader	きゅうごうりーだー	固有名詞	6
9	98Reader	きゅうはちりーだー	固有名詞	6
10	2000Reader	にせんりーだー	固有名詞	4
11	訓盲院	くんもういん	固有名詞	2
12	国際障害者年	こくさいしょうがいしゃねん	固有名詞	9
13	国リハ	こくりは	固有名詞	4
14	身体障害者福祉司	しんたいしょうがいしゃふくしし	固有名詞	7
15	視障者	ししょうしゃ	固有名詞	3
16	色変	しきへん	固有名詞	1
17	筑盲	つくもう	固有名詞	4
18	東京点字出版所	とうきょうてんじしゅつぱんじょ	固有名詞	5
19	東点	とうてん	固有名詞	3
20	日盲連	にちもうれん	固有名詞	3
21	日点委	にってんい	固有名詞	3
22	日点	にってん	固有名詞	3
23	盲ろう者	もうろうしゃ	固有名詞	3
24	盲	もう	固有名詞	2
25	low vision	ろーびじょん	固有名詞	3
26	旭電化工業	あさひでんかこうぎょう	固有名詞	7
27	芦森工業	あしもりこうぎょう	固有名詞	5
28	愛三工業	あいさんこうぎょう	固有名詞	5
29	安藤建設	あんどうけんせつ	固有名詞	5
30	飛島建設	あすかけんせつ	固有名詞	4
31	旭テック	あさひてっく	固有名詞	4
32	因幡電機産業	いなばでんきさんぎょう	固有名詞	7
33	稲畑産業	いなばたさんぎょう	固有名詞	5
34	伊藤忠食品	いとうちゅうしょくひん	固有名詞	6
35	伊予銀行	いよぎんこう	固有名詞	3
36	NTTデータ	えぬていーていーでーた	固有名詞	7
37	NECソフト	えぬいーしーそふと	固有名詞	7
38	江崎グリコ	えざきぐりこ	固有名詞	4
39	小田急電鉄	おだきゅうでんてつ	固有名詞	5
40	小田急建設	おだきゅうけんせつ	固有名詞	5
41	大林道路	おおばやしどうろ	固有名詞	6
42	大倉工業	おおくらこうぎょう	固有名詞	5
43	大隈豊和機械	おおくまほうわきかい	固有名詞	8
44	兼松日産農林	かねまつにっさんのうりん	固有名詞	9
45	勝村建設	かつむらけんせつ	固有名詞	5
46	加賀電子	かがでんし	固有名詞	3
47	花王	かおう	固有名詞	3
48	近畿日本鉄道	きんきにっぽんてつどう	固有名詞	8
49	杏林製薬	きょうりんせいやく	固有名詞	5
50	クラヤ三星堂	くらやさんせいどう	固有名詞	9

51	熊谷組	くまがやぐみ	固有名詞	6
52	京王電鉄	けいおうでんでつ	固有名詞	5
53	神戸製鋼所	こうべせいこうしょ	固有名詞	8
54	光世証券	こうせいしょうけん	固有名詞	5
55	駒井鉄工	こまいてっこう	固有名詞	4
56	広濟堂	こうさいどう	固有名詞	6
57	三晃金属工業	さんこうきんぞくこうぎょう	固有名詞	9
58	三機工業	さんきこうぎょう	固有名詞	4
59	相模鉄道	さがみてつどう	固有名詞	4
60	三和エレック	さんわえれっく	固有名詞	4
61	山九	さんきゅう	固有名詞	3
62	常磐興産	じょうばんこうさん	固有名詞	5
63	品川白煉瓦	しながわしろれんが	固有名詞	7
64	新光商事	しんこうしょうじ	固有名詞	5
65	新光証券	しんこうしょうけん	固有名詞	5
66	塩野義製薬	しおのぎせいやく	固有名詞	5
67	昭文社	しょうぶんしゃ	固有名詞	3
68	住友林業	すみともりんぎょう	固有名詞	5
69	住友電設	すみともでんせつ	固有名詞	5
70	住江織物	すみのえおりもの	固有名詞	5
71	住友建設	すみともけんせつ	固有名詞	5
72	住友精化	すみともせい化	固有名詞	5
73	積水化成成品工業	せきすいかせいひんこうぎょう	固有名詞	9
74	ゼリア新薬工業	ぜりあしんやくこうぎょう	固有名詞	8
75	世紀東急工業	せいきとうきゅうこうぎょう	固有名詞	8
76	西華産業	せいかさんぎょう	固有名詞	4
77	大和ハウス工業	だいわはうすこうぎょう	固有名詞	7
78	大日本土木	だいにっぽんどぼく	固有名詞	7
79	大和工商リース	だいわこうしょうりーす	固有名詞	8
80	大日本塗料	だいにっぽんとりょう	固有名詞	7
81	大日本製薬	だいにっぽんせいやく	固有名詞	7
82	大日本印刷	だいにっぽんいんさつ	固有名詞	7
83	段谷産業	だんだにさんぎょう	固有名詞	5
84	大東紡織	だいたうぼうしょく	固有名詞	5
85	第一屋製パン	だいいちやせいぱん	固有名詞	6
86	大陽東洋酸素	たいようとうようさんそ	固有名詞	9
87	大東銀行	だいたうぎんこう	固有名詞	5
88	太平洋セメント	たいへいようせめんと	固有名詞	8
89	大豊建設	だいほうけんせつ	固有名詞	5
90	大東建託	だいたうけんたく	固有名詞	5
91	第四銀行	だいしぎんこう	固有名詞	4
92	高崎三興	たかさきさんこう	固有名詞	5
93	高田機工	たかだきこう	固有名詞	4
94	大庄	だいしょう	固有名詞	3
95	台糖	たいとう	固有名詞	3
96	中外炉工業	ちゅうがいろこうぎょう	固有名詞	6
97	中国工業	ちゅうごくこうぎょう	固有名詞	5
98	中央発条	ちゅうおうはつじょう	固有名詞	6
99	中外製薬	ちゅうがいでんこう	固有名詞	5
100	中電工	ちゅうでんこう	固有名詞	3
101	地産トーカン	ちさんとーかん	固有名詞	4
102	蝶理	ちょうり	固有名詞	1
103	椿本興業	つばきもとうぎょう	固有名詞	6

104	富山化学工業	とやまかがくこうぎょう	固有名詞	7
105	東海旅客鉄道	とうかいりよかくてつどう	固有名詞	8
106	東亜道路工業	とうあどうろこうぎょう	固有名詞	7
107	東亜建設工業	とうあけんせつこうぎょう	固有名詞	8
108	東燃ゼネラル石油	とうねんぜんねらるせきゆ	固有名詞	9
109	東急コミュニティー	とうきゅうこみゆにていー	固有名詞	7
110	東急リバブル	とうきゅうりばぶる	固有名詞	6
111	豊田紡織	とよだぼうしょく	固有名詞	4
112	東京都競馬	とうきょうとけいば	固有名詞	6
113	藤和不動産	とうわふどうさん	固有名詞	5
114	東急観光	とうきゅうかんこう	固有名詞	5
115	東和銀行	とうわぎんこう	固有名詞	4
116	豊田工機	とよだこうき	固有名詞	4
117	東電通	とうでんつう	固有名詞	3
118	TOTO	とーとー	固有名詞	3
119	東リ	とうり	固有名詞	3
120	日本合成化学	にっぽんごうせいかかく	固有名詞	9
121	日本カーバイド工業	にっぽんかーばいどこうぎょう	固有名詞	9
122	日本コンクリート	にっぽんこんくりーと	固有名詞	9
123	日本空港ビルディング	にほんくうこうびるでいんぐ	固有名詞	8
124	東日本旅客鉄道	ひがしにほんりよかくてつどう	固有名詞	9
125	日本梱包運輸	にっぽんこんぼううんゆ	固有名詞	9
126	日本高周波鋼業	にっぽんこうしゅはこうぎょう	固有名詞	9
127	日本金属工業	にっぽんきんぞくこうぎょう	固有名詞	9
128	ニッセイ同和	にっせいどうわ	固有名詞	5
129	日本特殊陶業	にっぽんとくしゅとうぎょう	固有名詞	8
130	日本興亜	にっぽんこうあ	固有名詞	7
131	日本パーカライズン	にほんぱーからいじんぐ	固有名詞	7
132	西日本旅客鉄道	にしにほんりよかくてつどう	固有名詞	9
133	日本冶金工業	にっぽんやきんこうぎょう	固有名詞	8
134	日本電信電話	にっぽんでんしんでんわ	固有名詞	9
135	日東大都工業	にっとうだいとこうぎょう	固有名詞	8
136	日本テレビ放送網	にほんてれびほうそうもう	固有名詞	9
137	日本電気硝子	にっぽんでんきがらす	固有名詞	8
138	日本甜菜製糖	にっぽんでんさいせいとう	固有名詞	9
139	西日本鉄道	にしにっぽんてつどう	固有名詞	7
140	西日本銀行	にしにっぽんぎんこう	固有名詞	7
141	日本電話施設	にっぽんでんわしせつ	固有名詞	8
142	日本鑄鉄管	にっぽんちゅうてつかん	固有名詞	8
143	日本軽金属	にっぽんけいきんぞく	固有名詞	7
144	新潟鉄工所	にいがたてつこうしょ	固有名詞	9
145	日本加工製紙	にっぽんかこうせいし	固有名詞	8
146	日本エア・リキード	にほんえありきード	固有名詞	6
147	日本テクナート	にっぽんてくなーと	固有名詞	7
148	日本金属	にっぽんきんぞく	固有名詞	5
149	日本カーボン	にっぽんかーぼん	固有名詞	5
150	日本触媒	にほんしょくばい	固有名詞	4
151	日本橋梁	にほんきょうりょう	固有名詞	4
152	日本カーリット	にほんかーりっと	固有名詞	4

153	日本舗道	にっぽんほどう	固有名詞	5
154	日本ヒューム	にっぽんひゅーむ	固有名詞	5
155	日本道路	にっぽんどうろ	固有名詞	5
156	日本テレコム	にっぽんてれこむ	固有名詞	5
157	日本通運	にっぽんつううん	固有名詞	5
158	日本ゼオン	にっぽんぜおん	固有名詞	5
159	日本製粉	にっぽんせいふん	固有名詞	5
160	日本航空	にっぽんこうくう	固有名詞	5
161	日本工営	にっぽんこうえい	固有名詞	5
162	日本ケミファ	にっぽんけみふぁ	固有名詞	5
163	日本ケミコン	にっぽんけみこん	固有名詞	5
164	日本ガイシ	にっぽんがいし	固有名詞	5
165	日本フィルコン	にほんふいるこん	固有名詞	4
166	西松建設	にしまつけんせつ	固有名詞	5
167	日本精化	にっぽんせいしか	固有名詞	5
168	日本酸素	にっぽんさんそ	固有名詞	5
169	日本毛織	にっぽんけおり	固有名詞	5
170	日本化成	にっぽんかせい	固有名詞	5
171	日特建設	にっとくけんせつ	固有名詞	5
172	日東製網	にっとうせいもう	固有名詞	5
173	日東製粉	にっとうせいふん	固有名詞	5
174	日東精工	にっとうせいこう	固有名詞	5
175	日本フェルト	にほんふえると	固有名詞	4
176	日本農薬	にほんのうやく	固有名詞	4
177	日本曹達	にほんそーだ	固有名詞	4
178	日本コムシス	にほんこむしす	固有名詞	4
179	日本オラクル	にほんおらくる	固有名詞	4
180	日本油脂	にほんゆし	固有名詞	5
181	阪神電気鉄道	はんしんでんきてつどう	固有名詞	8
182	長谷工	はせこう	固有名詞	3
183	長谷川香料	はせがわこうりょう	固有名詞	5
184	東日本銀行	ひがしにっぽんぎんこう	固有名詞	8
185	日阪製作所	ひさかせいさくしよ	固有名詞	8
186	日立プラント	ひたちぷらんと	固有名詞	4
187	日立電線	ひたちでんせん	固有名詞	4
188	久光製薬	ひさみつせいやく	固有名詞	5
189	古河機械金属	ふるかわきかいきんぞく	固有名詞	8
190	富士紡績	ふじぼうせき	固有名詞	3
191	富士レビオ	ふじれびお	固有名詞	3
192	不二製油	ふじせいゆ	固有名詞	3
193	不二家	ふじや	固有名詞	2
194	北興化学工業	ほくこうかがくこうぎょう	固有名詞	8
195	豊和工業	ほうわこうぎょう	固有名詞	4
196	北海製罐	ほっかいせいかん	固有名詞	1
197	丸全昭和運輸	まるぜんしょうわうんゆ	固有名詞	8
198	松尾橋梁	まつおきょうりょう	固有名詞	4
199	真柄建設	まがらけんせつ	固有名詞	4
200	松屋フーズ	まつやふーず	固有名詞	4
201	丸栄	まるえい	固有名詞	2
202	三益半導体	みますはんどうたい	固有名詞	2
203	三井木材工業	みついもくざいこうぎょう	固有名詞	8
204	三井松島産業	みついまつしまさんぎょう	固有名詞	8
205	三井化学	みついかがく	固有名詞	4

206	三菱伸銅	みつびししんどう	固有名詞	5
207	三菱製鋼	みつびしせいこう	固有名詞	5
208	三菱樹脂	みつびしじゅし	固有名詞	5
209	三菱化学	みつびしかがく	固有名詞	5
210	三井鉱山	みついこうざん	固有名詞	5
211	三井金属	みついきんぞく	固有名詞	4
212	明光ナショナル証券	めいこうなしよなるしょうけん	固有名詞	9
213	名糖産業	めいとうさんぎょう	固有名詞	5
214	明和産業	めいわさんぎょう	固有名詞	4
215	森永乳業	もりながにゅうぎょう	固有名詞	5
216	森精機製作所	もりせいきせいさくじょ	固有名詞	10
217	養命酒製造	ようめいしゅせいぞう	固有名詞	6
218	横河ブリッジ	よこかわぶりっじ	固有名詞	5
219	米久	よねきゅう	固有名詞	2
220	菱電商事	りょうでんしょうじ	固有名詞	5
221	菱洋エレクトロ	りょうようえれくとろ	固有名詞	6
222	菱食	りょうしょく	固有名詞	4
223	若築建設	わかちくけんせつ	固有名詞	5