

### 3.3 動作機能のアルゴリズム

以下ではROM内のプログラムに従って各部の動作機能がどのように実行されていくかを説明する。

<イニシャライズ>

図3.3.1にイニシャライズルーチンのフローチャートを示す。

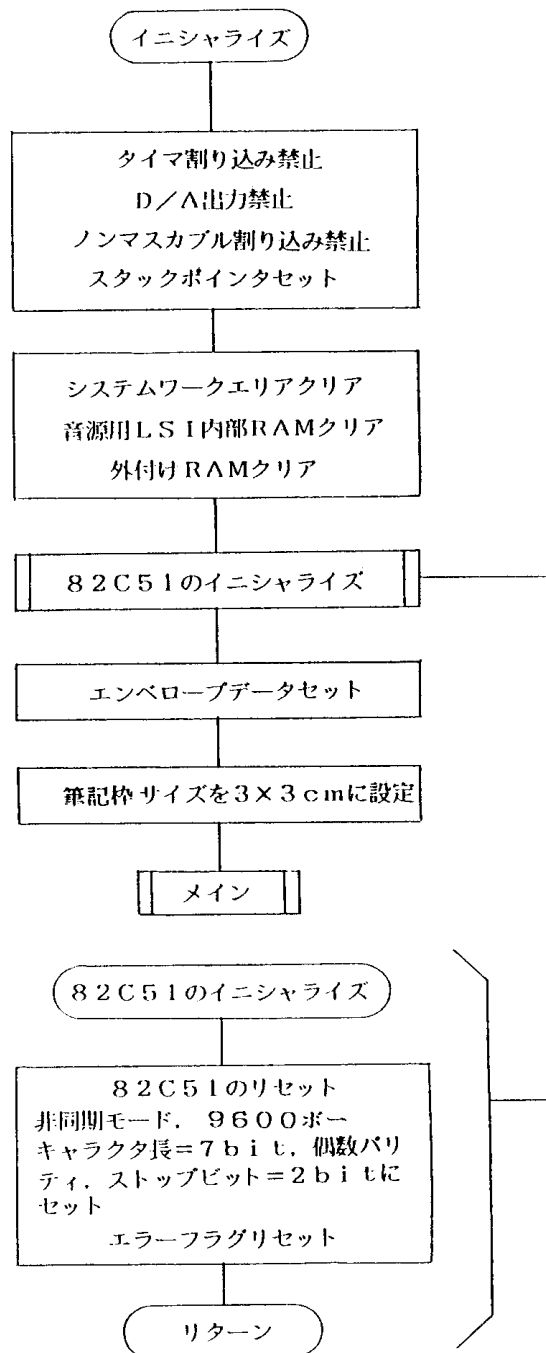


図3.3.1 イニシャライズルーチン

イニシャライズにあたって周期的タイマ割り込みを禁止し、D/Aの出力も禁止しておく。ノンマスカブル割り込みも禁止の状態にしておく。そして、スタックポインタをセットし、システムワークエリア、音源用LSIの内部RAM、外付けRAMをクリアする。その後、インターフェイス用LSIである82C51のイニシャライズを行ってから、音源LSI用のエンベロープ・データをセットする。

最後に筆記枠サイズを3×3cmに設定する。

<メインルーチン>

図3.3.2にメインルーチンのフローチャートを示す。

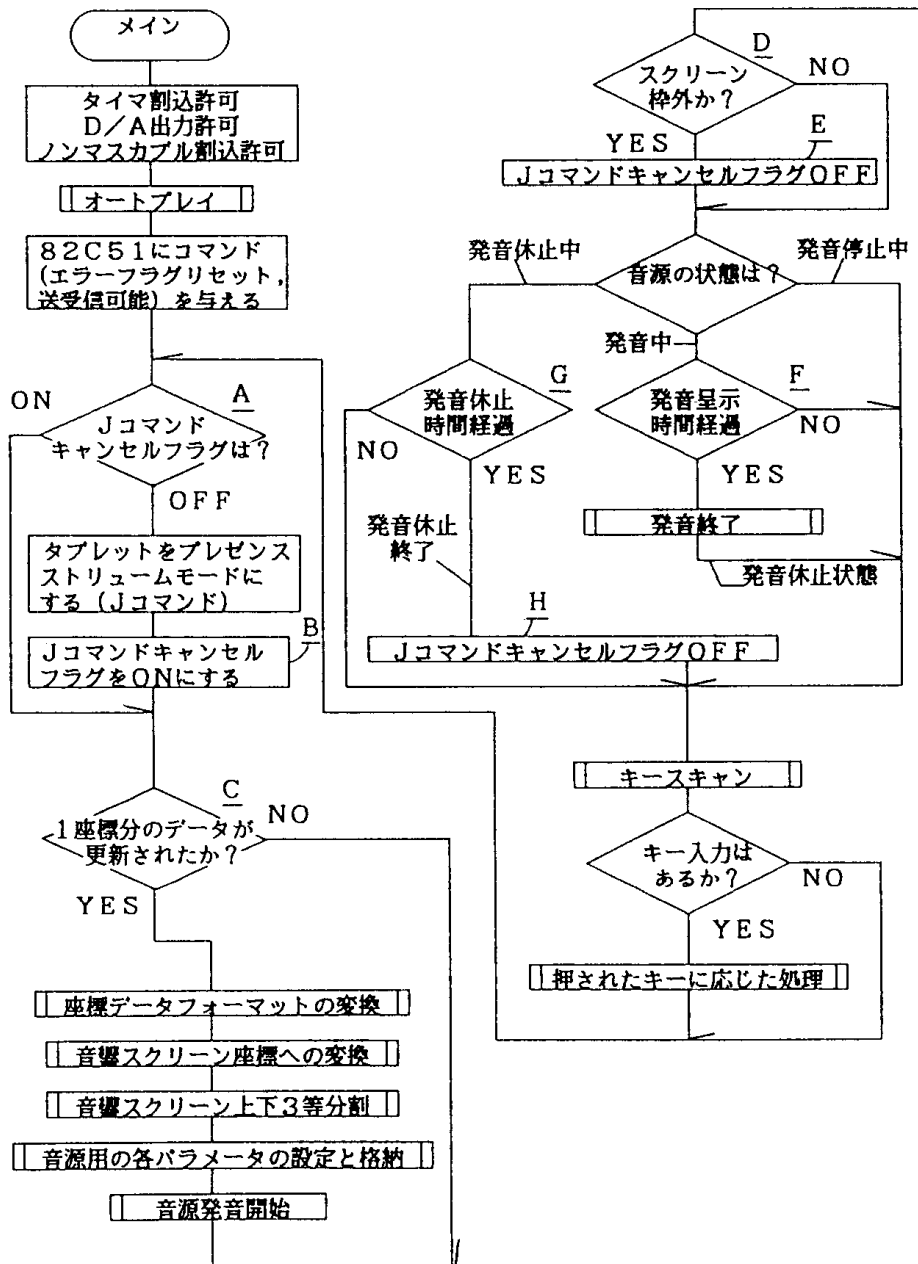


図3.3.2 メインルーチン

イニシャライズで禁止しておいたタイマ割り込み及びノンマスカブル割り込み、D/A出力を許可し、パワーオンを耳によって確認するための極短い自動演奏（<オートプレイ>）を行なった後、82C51にエラーフラグリセット、送受信可能にセットするためのコマンドを与える。

次にAでJコマンドキャンセルフラグを判別する。フラグがOFFの状態ならば、タブレットをプレゼンスストリウムモードにするためのJコマンドを与える。プレゼンスストリウムとはペンがタブレットにタッチしている間、連続して座標値を読みとるモードである。次段のBでこのフラグをONにする間のタイムラグを利用してタブレットからは4～5座標分のデータが送られてくる。既にこのルーチンを経てJコマンドキャンセルフラグがONとなっているならばAでONと判定されて何もしない。Bの次はCで送られてきた複数の座標の1番目のものを残すように座標を更新してYESと判定される。AでONと判定されれば何もしないのでCでNOと判定され即Dへ向う。Cで更新された座標データは処理しやすい形にするために<座標データフォーマットの変換>を行なう。フォーマット変換を行なったタブレットの絶対的座標データを音響スクリーン内での相対的座標データに変換する（<音響スクリーン座標への変換>）、そして音響スクリーンの上下方向の定位感を補助するために音色変化によって上下3等分割する（<音響スクリーンの上下3等分割>）、次に音源用LSIに与える各パラメータを設定してRAM領域へ一時的に格納しておく（<音源用の各パラメータの設定と格納>）、そして、RAMから各パラメータを音源LSIのシステムワークエリアに転送して音源を発音開始する（<音源発音開始>）、その後Dへ向かう。

Dでは現在ペンがタッチしているところが音響スクリーンの枠外かどうかを判別する。もし枠外だったならば、EにてJコマンドキャンセルフラグをOFFにして再び座標を更新する準備を行う。これによって音響スクリーン枠外をペンタッチしているときは、連続した音が聴こえてくる。なぜならAに戻る毎にOFFと判定されてJコマンドを絶えずタブレットに送ることになるため連続して座標を取得でき、常にCでYESと判定されて音源の発音をアクセスし続けるからである。

次に、音源がどのような状態にあるかを調べる。音源の状態としては、発音中、発音休止中（断続音を発生したときの間欠部分のこと）、発音停止中の3つの状態がある。ペンがスクリーン枠内にあるとき、発音中の場合は発音呈示時間が経過する迄は、EでNO AでON CでNO DでNOというループを繰返して前の発音状態を維持する。そして発音呈示時間が経過すればEでYESとなり<発音終了>の処理へ移り、発音休止状態となる。発音休止中の場合は発音休止時間が経過する迄はGでNO AでON CでNO DでNOというループで休止状態を維持する。そして発音休止時間が経過すればGでYESとなり発音休止終了となる。

発音休止が終了すれば再び発音を呈示しなければならないのでHでJコマンドキャンセルフラグをOFFにして座標を更新する準備を行う。これによりAでOFF CでYESと判定されて音源をアクセスする。ペンがタブレットにタッチしていないで発音停止中の場合は何もしない。

各スイッチの<キースキャン>では、キー入力があった場合は<押されたキーに応じた処理>を行なう。

以下ではメインルーチンで使用されるサブルーチンについて順次説明する。

### <オートプレイ>

図3.3.3にオートプレイルーチンのフローチャートを示す。

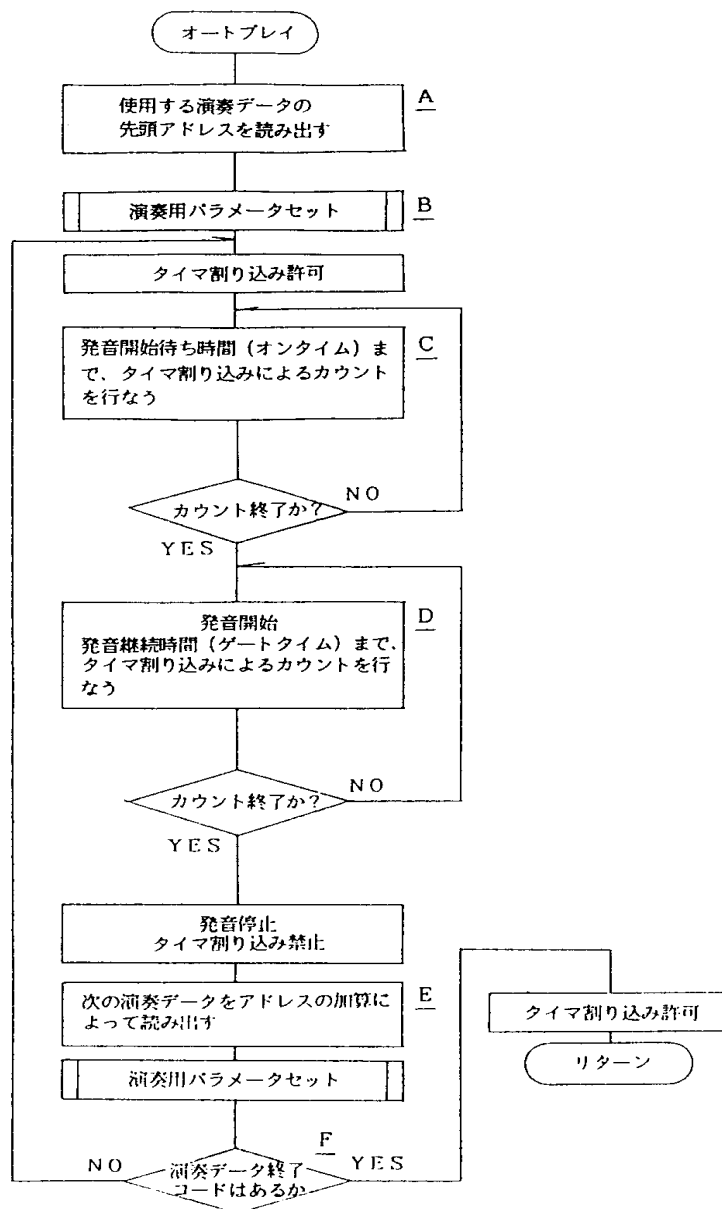


図3.3.3 オートプレイルーチン

パワーオン時や、各キースイッチが押されたときに盲人は耳で確認する必要がある。  
その時には自動的に極短い演奏を行なうようにした。

各演奏のデータフォーマット（表3.3.1参照）はノートコードとオクターブコードに1バイト、発音開始待ち時間（オンタイム）に1バイト、発音継続時間（ゲートタイム）に1バイト、発音ボリュームに1バイト使用している。

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
ノートコード				オクターブコード			
オンタイム（発音開始待ち時間）							
ゲートタイム（発音継続時間）							
発音ボリューム							

表3.3.1 イベントテーブルの演奏データフォーマット

まず、Aで使用する演奏データを格納するエリアの先頭アドレスを読み出す。そのアドレスを用いて第1音目の演奏データを読み出し、テンポラリーエリアに格納しておく。そして、Bでこの演奏データといっしょに指定の音色に対応したトーンデータをROM内のトーンテーブルから読み出して音源用LSIのシステムワークエリアに書き込む。

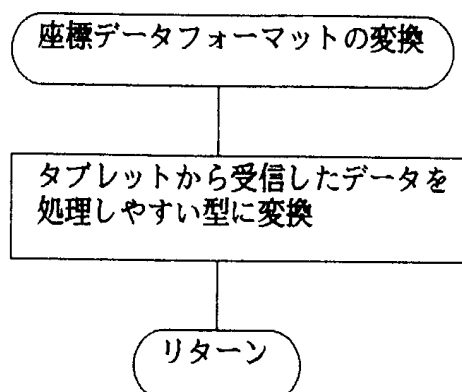
以上で第1音の演奏用のパラメータのセットは完了する。この後、タイマ割り込みを許可する。

Cにてタイマ割り込みによるカウントアップ動作を行い、発音開始待ち時間（オンタイム）が経つまで待つ。発音開始待ち時間が経過した時点で、発音を開始する。Dにて発音継続時間（ゲートタイム）になるまでタイマ割り込みによるカウントアップ動作を行い、それが終了した時点で発音を停止する。そしてタイマ割り込みを禁止する。これで第1音目の自動演奏が完了する。

次にEにて第2音の演奏データを読み込んでテンポラリーエリアに格納し、また前と同様に演奏用のパラメータのセットを行ない、C、Dにて発音の開始及び停止を行なう。こうして第2音、第3音……と自動演奏していく。Fにて読み込んだ演奏データが終了を示す場合はEの前で禁止していたタイマ割り込みを許可してリターンする。

<座標データフォーマットの変換>

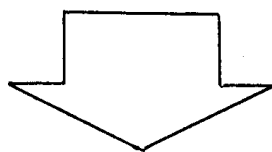
図3.3.4に座標データフォーマットの変換ルーチンのフローチャートを示す。



タブレットからの座標データフォーマット

	D7	D6	D5	D4	D3	D2	D1	D0
XDATA	P	0	X5	X4	X3	X2	X1	X0
XDATA+1	P	0	X11	X10	X9	X8	X7	X6
YDATA	P	0	Y5	Y4	Y3	Y2	Y1	Y0
YDATA+1	P	0	Y11	Y10	Y9	Y8	Y7	Y6

\*P ... パリティ



変換後の座標データフォーマット

	D7	D6	D5	D4	D3	D2	D1	D0
XDATA	X7	X6	X5	X4	X3	X2	X1	X0
XDATA+1	0	0	0	0	X11	X10	X9	X8
YDATA	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
YDATA+1	0	0	0	0	Y11	Y10	Y9	Y8

図3.3.4 座標データフォーマット変換ルーチン

J コマンドを受けてタブレットから送られてくる座標データはバイナリー形式のデータである。そのデータフォーマットを処理しやすい形にするためフォーマット変換を施す。

< 音響スクリーン座標への変換 >

図 3.3.5 にタブレットの座標と筆記枠の配置例を示す。

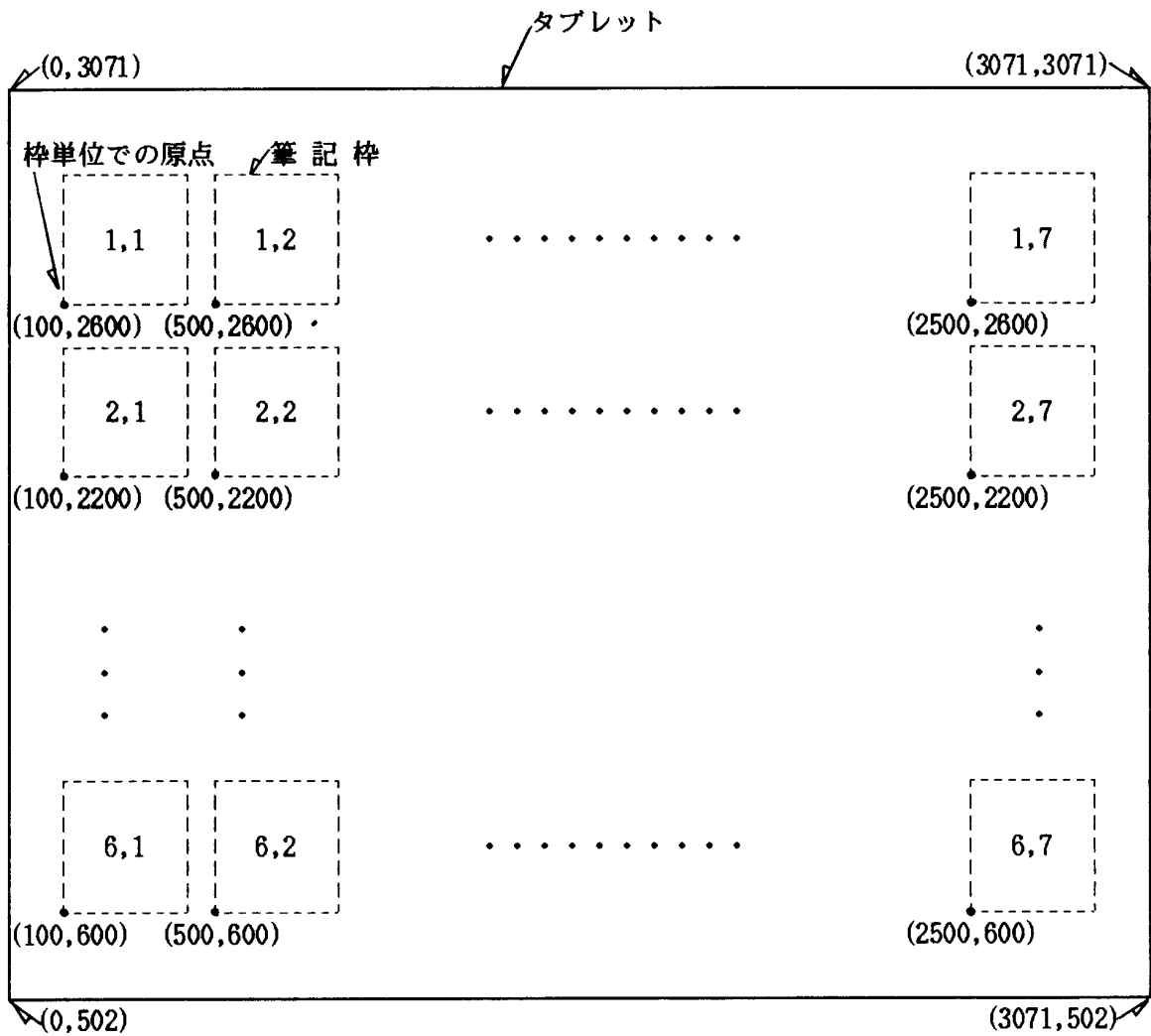


図 3.3.5 タブレットの座標と筆記枠サイズ 30×30mm (6 行×7 列) の配置例

筆記枠 30×30mm の例であるが、x、y 方向とも 400 ステップ (1 ステップ = 0.1mm に相応) 間隔で配置されている。

J コマンドを受けてタブレットから送られてくる (0, 502) ~ (3071, 3071) の範囲内の絶対的な座標データは、後述するように筆記枠の原点座標を基準にした相対的な座標 (音響スクリーン座標) に変換される。従って筆記を進めるときに筆記枠がかわれば原点座標を変更していただくだけの処理で済むことになる。

図 3.3.6 に音響スクリーン座標への変換方法を示す。

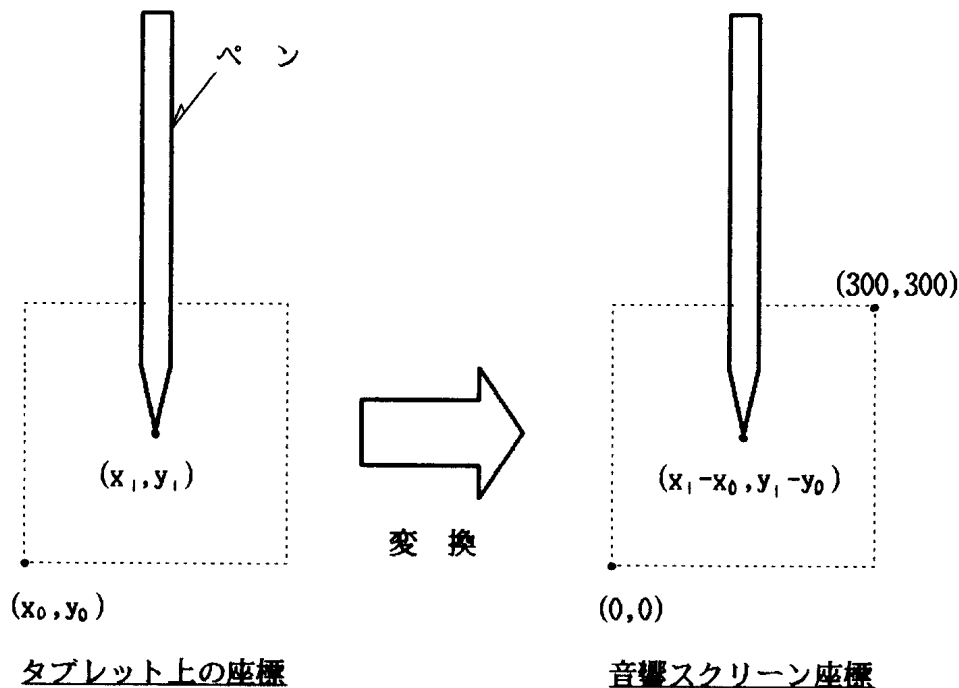


図 3.3.6 音響スクリーン座標への変換

30×30mm ならば 300×300 ステップの範囲となるのでタブレット上においてペンタッチした座標  $(x_1, y_1)$  から筆記枠の原点座標  $(x_0, y_0)$  を減算して音響スクリーン座標  $(0, 0) \sim (300, 300)$  内に変換する。この範囲外となった座標は枠外として判定されるが、負になった座標値は 0 に、300 より大きくなった座標値は 300 に設定し、枠の外周上に乗る座標データとする。

一方、音源用のパラメータテーブルは x 座標に対応して R L 音量バランスデータと発音呈示時間データが、y 座標に対応してメル尺度周波数分割データと発音呈示時間データが各々 301 ポイントづつ用意されている。(発音休止時間データは約 11msec 一定なので不要) そして変換した座標値をもとに各々から対応するデータを読み出してその座標における音源用のパラメータとする。従って枠外に出たときは枠外周上に対応したパラメータを使うことになる。枠サイズが変化してもこのパラメータテーブルが使われる。



例えば筆記枠サイズが  $30 \times 270\text{mm}$  のときは音響スクリーン座標は  $(0, 0) \sim (2700, 300)$  の範囲をとるが、パラメータテーブルを読み出すときはスクリーン枠サイズ変換テーブルによって  $(0, 0) \sim (300, 300)$  の範囲にまるめてしまう。従って枠サイズに関係なくパラメータテーブルは  $x, y$  方向とも 301 ポイント用意したものを共通に使用することになる。一方タブレットの座標は約  $0.1\text{mm}$  単位で出力されるが性能上の分解能は  $0.3\text{mm}$  である。従って、筆記枠の辺の長さが  $30\text{mm}$  だと実際上の座標の変化は  $30 / 0.3 = 100$  ステップで  $(0 \sim 100)$  の変化とみなせる。従って、パラメータテーブルからは 101 ポイントのデータを読み出すことになり、音響スクリーンにおける分解能は 101 ポイントとなる。辺長が  $30 \sim 90\text{mm}$  の分解能は 101 ~ 301 ポイントの分解能に対応し、 $90\text{mm}$  以上は 301 ポイントにまるめられる。

以上の一連の流れを図 3.3.7 にフローチャートとして示す。

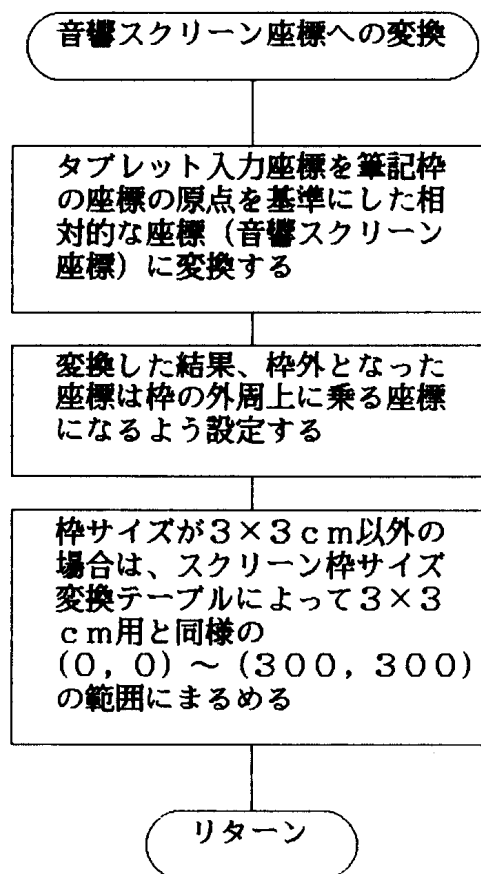


図 3.3.7 音響スクリーン座標変換ルーチン

#### < 音響スクリーンの上下 3 等分割 >

上下方向の筆記位置を探らせるために音響フィードバックは周波数の高低を利用しているが、全体的な位置を大ざっぱに把握することも必要なため、音響スクリーンの上下方向を 3 等分割してそれぞれに異なった音色を持たせている。

実際には、音響スクリーンの y 座標が、0 y 100 のとき音色 1 (フルート)、101 y 200 のとき音色 2 (クラリネット)、201 y 300 のとき音色 3 (ハーモニカ) という様に音色の指定を行なう。下からフルート、クラリネット、ハーモニカという順序になり、高周波が豊かな音色程上に配置して聴感的に上下感をだしている。

#### < 音源用の各パラメータの設定と格納 >

音響スクリーンの y 座標によって指定された音色のトーンデータをトーンテーブルから読み出す。読み出したトーンデータはパラメータ用テンポラリーエリアに格納する。そして前記したように (0, 0) ~ (300, 300) の範囲の音響スクリーン座標にもとづいて、x、y 各々に用意した 301 ポイントのパラメータテーブルから R L 音量バランスデータ、発音呈示時間データ、周波数データを読み出す。読み出したデータはトーンデータと同様にパラメータ用テンポラリーエリアに格納する。

#### < 音源発音開始 >

ここでは、パラメータ用テンポラリーエリアに格納された音源用の各パラメータを音源 L S I のシステムワークエリアに書き込む。そして発音開始フラグを ON にする。

#### < 発音終了 >

ペンタッチを行なっている最中では音源の発音終了の処理は断続音を作り出すために行なわれる。音源発音開始の後、一定時間 (発音呈示時間) が経ったとき、この発音終了の処理を行って一定時間 (発音休止時間) が経つまで発音の開始ができない様にする。また、ペンタッチを止めたときにも当然この発音終了の処理が行なわれるが、その場合においては発音休止状態が終われば音源を発音停止状態にする。

#### < キースキャン>、< 押されたキーに応じた処理 >

キースキャンは  $2 \times 4 = 8$  個のスイッチを単純にスキャンするルーチンである。キースキャンの結果、あるキーが押されていることを検出したならばそのキーに対する処理を行うが、詳細については 3.4 節を参照されたい。

以上がメインルーチンで使用するサブルーチンであり、メインルーチンと連係して処理を行なうことになる。