

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3671274号  
(P3671274)

(45) 発行日 平成17年7月13日(2005.7.13)

(24) 登録日 平成17年4月28日(2005.4.28)

(51) Int. Cl.<sup>7</sup>

G10H 1/00

F I

G10H 1/00

Z

請求項の数 2 (全 45 頁)

<p>(21) 出願番号 特願平10-360207                  (22) 出願日 平成10年12月18日(1998.12.18)                  (65) 公開番号 特開2000-181447(P2000-181447A)                  (43) 公開日 平成12年6月30日(2000.6.30)                  審査請求日 平成15年4月28日(2003.4.28)</p>	<p>(73) 特許権者 000001443                  カシオ計算機株式会社                  東京都渋谷区本町1丁目6番2号                  (74) 代理人 100074099                  弁理士 大菅 義之                  (72) 発明者 佐々木 博之                  東京都羽村市栄町3丁目2番1号 カシオ                  計算機株式会社羽村技術センター内                    審査官 板橋 通孝</p>
--	---

最終頁に続く

(54) 【発明の名称】 音楽情報送受信装置、受信装置及び記憶媒体

(57) 【特許請求の範囲】

【請求項1】

時間データの付加された音楽情報を送信可能な演奏装置、及び受信された音楽情報に基づいて再生を行う受信装置に接続され、

時間データが付加された音楽情報を記憶する記憶手段と、

現時刻をカウントする現時刻カウント手段と、

演奏時刻をカウントする演奏時刻カウント手段と、

前記記憶手段に記憶されている音楽情報を、当該音楽情報に付加された時間データと現時刻カウント手段でカウントされた現時刻とに基づいて決定されるタイミングで前記演奏装置に送信する第1の送信手段と、

前記演奏装置から送信された時間データの付加された音楽情報を受信する受信手段と、

前記受信手段にて受信された音楽情報を前記受信装置に送信する第2の送信手段と、

前記受信手段により最初に受信された音楽情報に付加された時間データに基づいて、前記演奏時刻カウント手段でカウントされる演奏時刻を変更する変更手段と、

前記記憶手段に記憶されている音楽情報を、当該音楽情報に付加された時間データと前記変更手段で変更された演奏時刻とに基づいて決定されるタイミングでかつ当該演奏時刻を付加して前記受信装置に送信する第3の送信手段と、

を具備したことを特徴とする送受信装置。

【請求項2】

時間データの付加された音楽情報を送信可能な演奏装置、及び受信された音楽情報に基

づいて再生を行う受信装置に接続された送受信装置に適用されるプログラムを記憶した記憶媒体であって、

現時刻をカウントするステップと、

演奏時刻をカウントするステップと、

記憶手段に記憶されている時間データが付加された音楽情報を、当該音楽情報に付加された時間データとカウントされた現時刻とに基づいて決定されるタイミングで前記演奏装置に送信するステップと、

前記演奏装置から送信された時間データの付加された音楽情報を受信するステップと、

前記受信された音楽情報を前記受信装置に送信するステップと、

前記最初に受信された音楽情報に付加された時間データに基づいて、前記カウントされた演奏時刻を変更するステップと、

前記記憶手段に記憶されている音楽情報を、当該音楽情報に付加された時間データと前記変更された演奏時刻とに基づいて決定されるタイミングでかつ当該演奏時刻を付加して前記受信装置に送信するステップと、

を有するプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、音楽情報の送受信により受信側で楽音を発音させて演奏を行わせるための技術に関する。

【0002】

【従来の技術および発明が解決しようとする課題】

現在では、MIDI (Musical Instrument Digital Interface) が普及したこともあって、音楽情報を送信し、受信側で楽音を発音させることが広く行われている。このことから、電子楽器や音源装置、或いはシンセサイザ等の楽音発生装置には、音楽情報送受信装置（或いは、音楽情報送信装置、及び音楽情報受信装置の一方）が幅広く搭載されている。

【0003】

上記音楽情報には、様々な種類がある。送信側と受信側とで実際にやりとりされる音楽情報の多くは、ノートオンやノートオフといったイベントのデータであり、そのデータ量は2、3バイトと比較的に小さい。しかし、楽音の波形データのように、大量のデータを1音楽情報として送信することもある。

【0004】

そのようなデータを音楽情報として送信する場合、送信を開始してから終了するまでにかかる送信時間が長くなる。その送信時間は、エラーの発生や送信経路の状況等の理由によって変動する。このため、波形データを送信して直ちに再生させるような場合、その送信時間によって再生を開始するタイミング（発音開始タイミング）が変化するという問題点があった。

【0005】

演奏（楽音の発音タイミング等）のズレは、その演奏から受ける印象を大きく変化させることが多く、音楽的な影響が大きい。そのため、従来は、波形データを前もって送信しておき、その波形データの再生を指示するコマンドを必要に応じて送信するといったように、波形データの使用方法には実質的に大きな制限があった。このことから、波形データをより有効に使用できるようにすることが強く要請されていた。

【0006】

ところで、MIDIの普及は、楽器の演奏スタイルや音楽制作方法、音楽のスタイルをも変化させている。楽音発生装置の利用方法の幅を広げて、これまでになかった音楽の楽しみ方を提供している。その利用方法の幅をより広げて、ユーザにとっての利便性を更に向上させるという要請も継続して存在していた。

【0007】

本発明の第1の課題は、音楽的な不具合の発生を回避しつつ、送受信される波形データの

10

20

30

40

50

使用方法をより広げることにある。

また、本発明の第2の課題は、音楽情報の送受信を通して楽音発生装置の利用方法の幅をより広げることにある。

【0008】

なお、本発明の参考技術文献としては、特願平10-102494号が挙げられる。

【0009】

【課題を解決するための手段】

本発明の音楽情報送受信装置は、時間データの付加された音楽情報を送信可能な演奏装置、及び受信された音楽情報に基づいて再生を行う受信装置に接続され、時間データが付加された音楽情報を記憶する記憶手段と、現時刻をカウントする現時刻カウント手段と、演奏時刻をカウントする演奏時刻カウント手段と、記憶手段に記憶されている音楽情報を、当該音楽情報に付加された時間データと現時刻カウント手段でカウントされた現時刻とに基づいて決定されるタイミングで演奏装置に送信する第1の送信手段と、演奏装置から送信された時間データの付加された音楽情報を受信する受信手段と、受信手段にて受信された音楽情報を受信装置に送信する第2の送信手段と、受信手段により最初に受信された音楽情報に付加された時間データに基づいて、演奏時刻カウント手段でカウントされる演奏時刻を変更する変更手段と、記憶手段に記憶されている音楽情報を、当該音楽情報に付加された時間データと変更手段で変更された演奏時刻とに基づいて決定されるタイミングでかつ当該演奏時刻を付加して前記受信装置に送信する第3の送信手段と、を具備する。

【0013】

また本発明の記録媒体は、時間データの付加された音楽情報を送信可能な演奏装置、及び受信された音楽情報に基づいて再生を行う受信装置に接続された送受信装置に適用されるプログラムを記憶した記憶媒体であって、現時刻をカウントするステップと、演奏時刻をカウントするステップと、記憶手段に記憶されている時間データが付加された音楽情報を、当該音楽情報に付加された時間データとカウントされた現時刻とに基づいて決定されるタイミングで演奏装置に送信するステップと、演奏装置から送信された時間データの付加された音楽情報を受信するステップと、受信された音楽情報を受信装置に送信するステップと、最初に受信された音楽情報に付加された時間データに基づいて、カウントされた演奏時刻を変更するステップと、記憶手段に記憶されている音楽情報を、当該音楽情報に付加された時間データと前記変更された演奏時刻とに基づいて決定されるタイミングでかつ当該演奏時刻を付加して前記受信装置に送信するステップと、を有するプログラムを記録している。

【0026】

本発明では、例えばパート別に音楽情報を分けて送信するといったように、複数の外部装置に送信する音楽情報の内容を外部装置単位で異ならせつつ、音楽情報をそれが有する時間データで指定されるタイミングに基づいて送信する。そのように、音楽情報の内容を異ならせられるようにすることにより、装置を使用するうえでの選択の幅が広がって利便性が向上し、ユーザは音楽情報を受信させる各装置をより所望する形で使用することが可能となる。

【0028】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態につき詳細に説明する。

<第1の実施の形態>

図1は、第1の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の回路構成図である。

【0029】

その楽音発生装置100は、図1に示すように、装置100全体の制御を行うCPU101と、時間を計時するタイマ(TIMER)102と、プログラムや演奏データ、及び各種制御用データ等を格納したROM103と、CPU101が作業用に用いるRAM104と、画面上に表示するカーソルの位置を指示するカーソルポインタ105と、そのポイ

10

20

30

40

50

ンタ105が指示する位置にカーソルを表示する液晶表示器106と、所定の送信経路を介して外部装置から送信された音楽情報を受信する受信器107と、その送信経路を介して外部装置に音楽情報を送信する送信器108と、CPU101から送られたデジタルの波形データをD/A変換するDAコンバータ109と、そのコンバータ109が出力したアナログの波形信号を音声に変換して出力するスピーカ110と、周囲の音を集音してアナログの音声信号を出力するマイク111と、そのマイク111が出力した音声信号をデジタルの波形データに変換するADコンバータ112と、音源装置120にCPU101から受け取ったMIDIデータを出力するMIDI OUT端子113と、鍵盤装置(キーボード)130から出力されたMIDIデータを入力するMIDI IN端子114と、各種スイッチを有するスイッチ群115と、を備えて構成されている。

10

#### 【0030】

上記送信経路は、例えば、LANやWAN、或いは公衆網等のネットワークである。本実施の形態による音楽情報送受信装置は、上述した構成を有する楽音発生装置100において、MIDI IN端子114、及びMIDI OUT端子113を介して外部装置との間でMIDIデータの送受信を行う一方、受信器107、及び送信器108を介してネットワークに接続されたノード(装置)との間で音楽情報の送受信を行う装置として実現されている。

#### 【0031】

ここでは、便宜的に、受信器107、及び送信器108は送信経路としてLANを介して他の装置とデータの受信、或いは送信を行うことを前提とすることにする。その前提では、受信器107と送信器108は一つの装置として存在することになるが、データの受信と送信の区別を明確にするために、それらの用語はそのまま用いることにする。LANを介してデータを送受信する方法には、幾つかあるが、便宜的に、ここでは特に図示しないサーバを介してデータを送受信することを前提とすることにする。

20

#### 【0032】

なお、MIDI OUT端子113、或いはMIDI IN端子114に接続させる外部装置は、音源装置120、或いは鍵盤装置130に限定されるものではない。自動演奏装置や電子楽器、或いはコントローラ等のMIDI端子を有する装置と任意に接続させることができる。

#### 【0033】

上記の構成において、その動作を説明する。  
不図示の電源がオンされると、CPU101はROM103に格納されたプログラムを読み出して実行することにより、装置100全体の制御を開始する。このとき、例えばタイマ102を起動させて、時間の計時を開始させる。その後は、RAM104を作業用として使用しながら、スイッチ群115、MIDI IN端子115(鍵盤装置130)、ADコンバータ112、及び受信器107の各入力装置により得られた情報に応じた制御を行う。

30

#### 【0034】

スイッチ群115は、特に詳細な説明は省略するが、例えば液晶表示器106に表示させる情報の種類や音色、或いは送信先等を指定できるようにするためのスイッチ(以降、種類選択スイッチと呼ぶ)、送信先や音色、液晶表示器106に楽譜として表示させる曲等を指定するための複数のボタン(以降、便宜的に汎用ボタン群と呼ぶ)等の各種スイッチと、それらの操作状態を検出する検出回路と、から構成されたものである。その検出回路は、例えば、CPU101の指示に従ってそれらのスイッチの走査(スキャン)を行い、その走査結果を操作情報としてCPU101に送出する。

40

#### 【0035】

CPU101は、スイッチ群115から送られた操作情報の解析を行い、ユーザが操作したスイッチ、及びその操作内容を特定する。それにより、ユーザの指示内容を認識し、その認識結果に応じて各種の設定を行い、その設定に従って装置100全体の制御を行う。例えば楽譜の表示を指示された場合には、その画像データをROM103から読み出し、

50

それを液晶表示器106が備えたVRAM(図示せず)に転送することにより、液晶表示器106にユーザが指定した楽譜を表示させる。種類選択スイッチで送信先の指定を選択した後、汎用ボタン群をユーザが操作した場合には、汎用ボタン群への操作に応じて送信先を設定する。なお、各種の設定は、通常、その種類別に用意した変数の値を書き換えることで行われる。

**【0036】**

鍵盤装置130は、搭載された鍵盤や各種スイッチに対してユーザが行った操作内容を、MIDIデータの形で出力する。CPU101は、MIDI IN端子114を介してそのMIDIデータを受け取ると、それをMIDI OUT端子113に出力することにより、音源装置120に楽音の発音、或いは消音を行わせる。音楽情報の送信先が指定されている場合には、MIDI IN端子114を介して受信したMIDIデータに、MIDIデータを送信することを表す識別子、及びそれを処理すべきタイミングを表す時間データを付加して送信器108から音楽情報として送信する。その時間データは、例えばタイマ102の値を参照して更新する変数(現在時刻レジスタ)の値である。

10

**【0037】**

特に図示しないジャック等の端子にマイク111が接続されている場合、ADコンバータ112はマイク111から出力される音声信号をサンプリングして、デジタルの波形データを出力する。

**【0038】**

CPU101は、ADコンバータ112から受け取った波形データをDAコンバータ109に出力する。それにより、マイク111が集音した音をリアルタイムでスピーカ110から発音させる。送信先が指定されている場合には、その波形データを格納する2つの領域をRAM104に確保して、それらの領域に波形データを交互に格納(PCM録音)しつつ、そのうちの一方に格納した波形データを順次、送信器108から音楽情報として送信する。波形データを送信することを表す識別子は、それを送信する際に付加する。

20

**【0039】**

本実施の形態では、上述したようにして、波形データは確保した領域の容量を単位として分割し送信している。これは、波形データの送信を間欠的にし、その送信間隔の間に他の音楽情報(例えばMIDIデータ)の送信を行えるようにするためである。言い換えれば、どのような送信経路を用いたとしても、異なる種類の音楽情報を並行して送信できるようにするためである。

30

**【0040】**

なお、波形データの格納(及び送信)用に確保した2つの領域については、以降、便宜的に波形バッファA、波形バッファBと呼ぶことにする。それらを明確に区別する必要がないときには、単に波形バッファと呼ぶことにする。本実施の形態では、波形バッファの容量を256バイトとしている。

**【0041】**

その波形バッファの容量単位で波形データを送信する度に、その再生を指示するコマンド(以降、再生指示コマンドと呼ぶ)を送信する。受信側が波形データを正常に受信できていない場合、結果としてノイズを再生してしまう恐れがある。このことから、そのコマンドには、それまでに送信したデータ量が正常に受信できたか否かを受信側が判別するためのデータを付加して送信している。更には、その再生を開始すべきタイミングを表す時間データを付加している。それらのデータを付加して、送信器108から音楽情報として送信している。なお、その時間データは、例えばタイマ102の値を基に更新する変数(録音時刻レジスタ)の値である。

40

**【0042】**

再生指示コマンドに時間データを付加して波形データを再生するタイミングを送信先に指定することにより、たとえ波形データ毎に送信時間が変動しても、受信側でその変動を吸収することができる。それにより、送信時間に変動が生じる送信路を用いても波形データを受信側がより適切なタイミングで再生することができることから、送信路や状況等によ

50

る音楽的な不具合の発生が回避され、波形データを演奏に適切に用いることができる。従って、波形データの使用方法の幅も広がることになる。

【0043】

ところで、波形データを送信する場合、受信側が発音できる音の制限がなくなり、様々な音を鳴らせるようになる。このため、音楽表現の幅が広がり、新しい音楽スタイルの創造にも寄与することができるという効果もある。

【0044】

ここで、上述したようにして送信される音楽情報のデータ構成について説明する。ここでは、便宜的に、音楽情報として、MIDIデータ、波形データ、再生指示コマンド、楽譜の画像(ビットマップ)データ、及びその楽譜上における演奏位置を示すポイントデータ、の計4種類の音楽情報を送信するものとして説明することにする。

10

【0045】

上記MIDIデータは、例えばエクスクルーシブ・メッセージを除いたものである。そのメッセージを除くMIDIデータは、1~3バイトの可変長のデータである。しかし、本実施の形態では、処理を簡単にするため、MIDIデータを4バイトの固定長のデータとして扱っている。それに付加する時間データは、例えば時間をms単位で表す4バイトのデータとしている。従って、MIDIデータを音楽情報として送信する場合、そのデータ量は計8バイトとなる。

【0046】

それ以外のデータを音楽情報として送信する場合、上記MIDIデータと時間データの代わりに各々1バイトが割り当てられる。それら(計2バイトのデータ)は、音楽情報の種類を表す識別子(送信先に対するコマンドに相当する)を格納するためのものである。その識別子の内容を(1バイト目の値, 2バイト目の値)の形で表現すると、(0, 0)は波形データを表し、(1, 0)は再生指示コマンド、(2, 0)は画像データ、(3, 0)はポイントデータをそれぞれ表している。これらの識別子が付加される音楽情報のデータ構成は、以下のようになっている。

20

【0047】

先ず、識別子(0, 0)が割り当てられた波形データでは、その識別子に、4バイトを割り当てた波形データのバイト数を表すデータ、同じく4バイトを割り当てた波形データのシリアル番号を表すデータ、が続き、その後に波形データが付加される。ここでは、音楽情報を構成するデータの単位(識別子や波形データのバイト数を表すデータ等)をワードと呼ぶことにする。

30

【0048】

識別子(1, 0)が割り当てられた再生指示コマンドでは、その識別子自体が再生指示コマンドを表している。そのコマンドを兼ねた識別子に、4バイトを割り当てた波形データの総バイト数を表すデータ、同じく4バイトを割り当てた再生のタイミングを指定する時間データが付加される。2番目のワードが、それまでも含めて波形データが正常に受信できたか否かを送信先が判別するためのデータである。

【0049】

識別子(2, 0)が割り当てられた画像データでは、その識別子に、その画像データが付加される。時間データは付加しない。これは、画像データとして楽譜を送信することを想定しており、その楽譜は演奏が始まる前に送信しておく必要があるためである。

40

【0050】

識別子(3, 0)が割り当てられたポイントデータでは、その識別子に、4バイトを割り当てたX座標上の位置を指定するX座標データ、同じく4バイトを割り当てたY座標上の位置を指定するY座標データが付加される。ここでも、時間データは付加しない。これは、楽音の発音と比較して、画像データの表示の切り換え等に要求される時間的精度は低いためである。

【0051】

CPU101は、音楽情報を、上記した形式で送信器108から送信させる。そのような

50

音楽情報は、他のノードで楽音発生装置 100 が送信先として指定された場合、受信器 107 によって受信される。

【0052】

なお、音楽情報は、実際には、送信先や送信元等の各種送信に係わる制御データが格納されたヘッダが付加して送信される。その送信された音楽情報は、サーバに保持される。その音楽情報の送信先は、サーバに自分宛に送信されたデータがあるか否かを確認する問い合わせを行う。それによって新たに送信されたデータがあることが判明した場合に、サーバからダウンロードすることで取得する。このように、音楽情報はサーバを介してやりとりされる。

【0053】

受信器 107 は、音楽情報を受信すると、その旨を CPU 101 に通知した後、受信した音楽情報を CPU 101 に送る。

CPU 101 は、音楽情報を受信器 107 から受け取ると、その先頭に位置する 2 バイトの値から音楽情報の種類を判別する。その後は、以下のように、判別結果に従って音楽情報の処理を行う。

【0054】

なお、MIDI データの場合、1 バイト目のデータの値は F0H (H は 16 進数での表現であることを表す) 以上であり、識別子は 2 バイトのデータである。このことから明らかのように、先頭の 2 バイトの値から音楽情報の種類を判別することができる。

【0055】

音楽情報が MIDI データと判別した場合、CPU 101 は、その MIDI データを、それに付加された時間データで指定されるタイミングで MIDI OUT 端子を介して音源装置 120 に送出する。

【0056】

波形データを送信する場合と同様に、受信した波形データを格納する 2 つの領域を RAM 104 に確保する。音楽情報が波形データと判別した場合には、それらの領域に、波形データを受信する度に交互に切り替えながらそれを格納する。その一方では、それまでに受信した波形データの総バイト数を算出し、その算出結果を変数 HSSR に代入する。なお、それらの領域については、以降、受信波形バッファ A、受信波形バッファ B と呼ぶことにする。それらを区別する必要がないときには、単に受信波形バッファと呼ぶことにする。本実施の形態では、受信波形バッファの容量を波形バッファと同じく、256 バイトとしている。

【0057】

音楽情報が再生指示コマンドと判別した場合には、その音楽情報中で先頭から 2 番目のワードの値と上記変数 HSSR の値とを比較し、それらが一致するか否か判定する。それまでに送信された波形データを正常に受信していたときには、それらの値は一致する。このことから、それらの値が一致していれば、音楽情報中の時間データで指定されるタイミングとなるのを待って、受信波形バッファに格納した波形データの再生を開始する。反対に一致していなければ、波形データの受信にエラーが発生しているとして、波形データの再生を終了する。それにより、演奏を損なわせる音の再生を回避する。なお、波形データの再生は、CPU 101 が、受信波形バッファから 1 サンプリング分の波形データを、サンプリング周期毎に順次読み出して DA コンバータ 109 に送出することで行われる。

【0058】

音楽情報が画像データと判別した場合には、そのことを示す識別子に続く画像データを液晶表示器 106 が備えた VRAM (図示せず) に格納する。カーソルポインタ 105 には、画像データを先頭から表示することを示す初期値を設定する。それにより、液晶表示器 106 の画面上に、画像データとして送られた楽譜の先頭部分を表示させる。

【0059】

特に詳細な説明は省略するが、その画像データは、ROM 103 に複数格納されており、スイッチ群 115 を介してユーザが選択するようになっている。それにより、CPU 10

10

20

30

40

50

1 は、ユーザが選択した画像データを音楽情報として送信する。

【 0 0 6 0 】

上記カーソルポインタ 1 0 5 は、カーソル（図示せず）の表示位置を X Y 座標上の位置で指定する。X 座標上の位置を示す値を格納したレジスタをカーソル X レジスタ、Y 座標上の位置を示す値を格納したレジスタをカーソル Y レジスタと呼ぶことにすると、それらの値を、演奏の進行に応じて更新する。その更新は、例えば、液晶表示器 1 0 6 に表示された楽譜（演奏データ）の内容、タイマ 1 0 2 が計時した時間、設定されているテンポ値等を基に行う。それにより、演奏の進行に合わせて、カーソルを表示させる楽譜上の位置を移動させ、楽譜の表示内容を更新（例えばスクロール）させる。なお、上記のレジスタは、当然のことながら、変数であっても良い。

10

【 0 0 6 1 】

音楽情報として画像データを受信した場合には、CPU 1 0 1 は、ポイントデータと判別される音楽情報が受信される度に、音楽情報中の 2 ワード目のデータをカーソル X レジスタに格納し、3 ワード目のデータをカーソル Y レジスタに格納する。それにより、液晶表示器 1 0 6 の画面上に表示させたカーソルの位置を、受信されたポイントデータが指定する位置に移動させる。

【 0 0 6 2 】

次に、上述した制御を行う CPU 1 0 1 の動作について、図 2 ~ 図 1 2 に示す各種動作フローチャートを参照して詳細に説明する。本発明は音楽情報の送受信に特に係わるものであることから、ここでは、理解を容易にするために、音楽情報の送信、及びその受信に着目して説明することにする。

20

【 0 0 6 3 】

図 2 は、音楽情報送信処理の動作フローチャートである。音楽情報の送信に着目して、その処理の流れを抜粋して示したものである。ここでは、便宜的に、音楽情報として、波形データ、画像データ（楽譜）、及びポイントデータを送信することを前提としている。始めに、図 2、更には図 3、及び図 4 を参照して、音楽情報を送信する際の CPU 1 0 1 の動作について詳細に説明する。

【 0 0 6 4 】

なお、図 2 に示す音楽情報送信処理（図 3 及び図 4 のそのサブルーチン処理を含む）は、ユーザがスイッチ群 1 1 5 を操作して送信先を指定した場合に、CPU 1 0 1 が ROM 1 0 3 に格納されているプログラムを実行することで実現される。

30

【 0 0 6 5 】

まず、ステップ 2 0 1 では、変数 F X、及び F Y にそれぞれ 0 を代入する。続くステップ 2 0 2 では、変数 X、Y にそれぞれ 0 を代入する。それが終了した後、ステップ 2 0 3 に移行する。

【 0 0 6 6 】

上記変数 F X、及び F Y は、カーソルの表示位置を CPU 1 0 1 が管理するための変数である。変数 X、及び Y は、それらの変数 F X、F Y を更新する作業を行うために用いられる変数である。

【 0 0 6 7 】

ステップ 2 0 3 では、2 バイトの定数（2, 0）（2 は 1 バイト目の値、0 は 2 バイト目の値である。以降、同様）を送信させる。その定数（2, 0）は、音楽情報として画像データ（本実施の形態では楽譜のビットマップデータ）を送信することを表す識別子である。

40

【 0 0 6 8 】

ステップ 2 0 3 に続くステップ 2 0 4 では、ユーザがスイッチ群 1 1 5 を介して選択した画像データを ROM 1 0 3 から読み出し、それを送信器 1 0 8 に送って送信させる。次のステップ 2 0 5 では、その画像データの送信が終了したか否か判定する。その送信が終了していない場合、判定は NO となり、ステップ 2 0 4 に戻って送信を継続して行わせる。そうでない場合には、判定は YES となってステップ 2 0 6 に移行する。

50

## 【 0 0 6 9 】

このようにして、ステップ 2 0 1 ~ 2 0 5 では、カーソルの表示位置の初期設定に係わる処理、及び画像データの送信に係わる処理が行われる。なお、本実施の形態では、ステップ 2 0 4 で送信する画像データを液晶表示器 1 0 6 の V R A M に転送して、その画面上に楽譜を表示するようにしている。これは、その楽譜を見て行ったユーザの演奏内容を音楽情報として送信することを想定しているためである。

## 【 0 0 7 0 】

ステップ 2 0 6 では、分割して送信する波形データの送信回数をカウントするための変数 S C に 0 を代入する。続くステップ 2 0 7 では、送信した波形データの総バイト数を保持しておくための変数 B C に 0 を代入する。それが終了すると、ステップ 2 0 8 に移行して、波形データの格納先を管理するための変数 R H B F に、波形バッファ A を示す値を代入する。その値を代入した後、ステップ 2 0 9 に移行する。

10

## 【 0 0 7 1 】

なお、変数 R H B F に代入する値は、便宜的に、波形バッファ A を示す値のときには A、波形バッファ B を示す値のときには B と表現することにする。波形バッファ、及び受信波形バッファに係わる他の変数においても同様に表現することにする。

ステップ 2 0 9 では、A D コンバータ 1 1 2 から出力された波形データを入力する。続くステップ 2 1 0 では、その波形データを、変数 R H B F によって指定される波形バッファに格納する。その格納を行った後、ステップ 2 1 1 に移行する。なお、A D コンバータ 1 1 2 はマイク 1 1 1 が出力する音声信号を A D 変換して出力することから、以降、波形データは音声データと呼ぶことにする。

20

## 【 0 0 7 2 】

ステップ 2 1 1 では、状況の変化に応じて音楽情報を送信するラベル 1 処理を実行する。続くステップ 2 1 2 では、音楽情報の送信の終了が指示されたか否か判定する。スイッチ群 1 1 5 の所定のスイッチを操作して音楽情報の送信の終了をユーザが指示した場合、判定は Y E S となって一連の処理を終了する。そうでない場合には、判定は N O となってステップ 2 1 1 に戻る。それにより、ユーザが送信の終了を指示するまで、ステップ 2 1 1 のラベル 1 処理を継続して実行する。

## 【 0 0 7 3 】

上述のステップ 2 0 1 ~ 2 1 0 は、音楽情報の送信を開始するうえでの初期設定を行うための処理、或いはその開始に伴う処理として実行される。M I D I データの受信、音声データを格納している波形バッファの空容量が無くなったといった状況の変化はステップ 2 1 1 のラベル 1 処理の実行によって対処される。

30

## 【 0 0 7 4 】

図 3 は、上記ステップ 2 1 1 として実行されるラベル 1 処理の動作フローチャートである。次に、図 3 を参照して、上記音楽情報送信処理中で実行されるサブルーチン処理について説明する。

## 【 0 0 7 5 】

上述したように、時間データは m s 単位でタイミングを表すデータとしている。タイマ 1 0 2 は、それよりも細かい単位で時間の計時を行う。このことから、本実施の形態では、現在時刻を保持させる変数を用意し、その変数の値を、タイマ 1 0 2 による時間の計時の進行に応じて更新するようにしている。その変数については、以降、現在時刻レジスタと呼ぶことにする。

40

## 【 0 0 7 6 】

まず、ステップ 3 0 1 では、現在時刻レジスタの値を 1 m s が経過する毎に更新するために、そのレジスタの値を更新してから 1 m s が経過したか否か判定する。現在のタイマ 1 0 2 の値が、現在時刻レジスタの値を前回、更新したときの値から 1 m s に対応する値以上大きくなっていった場合、判定は Y E S となり、ステップ 3 0 2 に移行してそのレジスタの値のインクリメントを行った後、ステップ 3 0 3 に移行する。そうでない場合には、判定は N O となってステップ 3 0 6 に移行する。

50

## 【 0 0 7 7 】

ステップ 3 0 3 では、M I D I I N 端子 1 1 4 が M I D I データを受信したか否か判定する。ユーザが鍵盤装置 1 3 0 を操作した場合、鍵盤装置 1 3 0 は M I D I データを出力することから、判定は Y E S となってステップ 3 0 4 に移行する。そうでない場合には、即ちユーザが鍵盤装置 1 3 0 の操作を行っていないような場合には、判定は N O となってステップ 3 0 6 に移行する。

## 【 0 0 7 8 】

ステップ 3 0 4 では、M I D I I N 端子 1 1 4 が受信した M I D I データを送信器 1 0 8 に送って送信させる。続くステップ 3 0 5 では、現在時刻レジスタの現在値を送信器 1 0 8 に送り、それを時間データとして送信させる。その後、ステップ 3 0 6 に移行する。なお、実際には、M I D I データ、及び時間データは、一つの packets として送信器 1 0 8 から送信される。

10

## 【 0 0 7 9 】

ステップ 3 0 6 では、カーソルポインタ 1 0 2 のカーソル X レジスタ、及びカーソル Y レジスタの各値を読み出し、それらを変数 X、及び Y にそれぞれ対応させて代入する。続くステップ 3 0 7 では、変数 F X の値が変数 X の値と等しく、且つ変数 F Y の値が変数 Y の値と等しいか否か判定する。カーソルポインタ 1 0 2 が指示するカーソルの表示位置がそれまでから変化した場合、変数 F X の値と変数 X の値、或いは変数 F Y の値と変数 Y の値が異なることから、判定は Y E S となってステップ 3 0 8 に移行する。そうでない場合には、判定は N O となってステップ 3 1 1 に移行する。

20

## 【 0 0 8 0 】

カーソルの表示位置が変化したことは、受信側のカーソルの表示位置を変更させる必要が生じたことを意味する。このことから、ステップ 3 0 8 ~ 3 1 0 では、ポイントデータを音楽情報として送信するための処理が行われる。

## 【 0 0 8 1 】

まず、ステップ 3 0 8 では、変数 F X に変数 X の値を代入し、変数 F Y に変数 Y の値を代入することにより、変数 F X、及び F Y の値の更新を行う。続くステップ 3 0 9 では、ポイントデータを音楽情報として送信することを表す定数 ( 3 , 0 ) を送信器 1 0 8 に送信させる。それを送信させると、ステップ 3 1 0 に移行して、変数 X、及び Y の値を送信器 1 0 8 に送信させる。それら変数の値を送信した後、ステップ 3 1 1 に移行する。なお、それら識別子、及び変数は、実際には 1 つの packets で送信器 1 0 8 から送信される。

30

## 【 0 0 8 2 】

ステップ 3 1 1 以降では、波形バッファへの音声データの格納やその波形バッファの切り換え、及び波形バッファに格納させた音声データの送信に係わる処理が行われる。

## 【 0 0 8 3 】

まず、ステップ 3 1 1 では、波形バッファの容量単位の録音が終了したか否か判定する。変数 R H B F の値が指定する波形バッファの空容量が無くなった場合、判定は Y E S となってステップ 3 1 2 に移行し、そうでない場合には、判定は N O となってステップ 3 1 6 に移行する。

## 【 0 0 8 4 】

ステップ 3 1 2 では、録音時刻レジスタに現在時刻レジスタの値を代入する。続くステップ 3 1 3 では、送信する音声データを格納した波形バッファを管理するための変数 S H B F に、変数 R H B F の値を代入する。その後は、ステップ 3 1 4 に移行する。

40

## 【 0 0 8 5 】

ステップ 3 1 4 では、変数 R H B F の値を反転させる。具体的には、それまでの値が A であれば B に、それまでの値が B であれば A に書き換える。その書き換えを終了すると、ステップ 3 1 5 に移行して、音声データを含む音楽情報の送信を行うラベル 2 処理を実行する。その実行を終了した後は、ステップ 3 1 6 に移行する。

## 【 0 0 8 6 】

ステップ 3 1 6 では、音声データを入力するタイミングとなったか否か判定する。C P U

50

101は、予め定められたサンプリング周期でA/Dコンバータ112が出力する音声データを取り込む。このことから、前回A/Dコンバータ112の出力を取り込んでから1サンプリング周期に対応する時間が経過した場合、判定はYESとなってステップ317に移行する。そうでない場合には、判定はNOとなって一連の処理を終了する。

【0087】

このようにして、本実施の形態では、波形バッファの容量単位の録音が終了する度に、音声データを格納する波形バッファを切り換えてその録音（格納）を継続して行うようにしている。切り換え前の波形バッファ、即ちそれまで音声データの格納を行っていた波形データに格納された音声データは、後述するラベル2処理の実行時に送信するようにしている。それにより、音声データを格納する波形バッファと音声データを送信する波形バッファとが一致するようなことを回避している。

10

【0088】

図4は、上記ステップ315として実行されるラベル2処理の動作フローチャートである。次に、図4を参照して、ラベル2処理について詳細に説明する。なお、そのラベル2処理には、上記ラベル1処理から現在時刻レジスタや録音時刻レジスタ等が引数として渡される。

【0089】

先ず、ステップ401では、現在時刻レジスタの値を1msが経過する毎に更新するために、そのレジスタの値を更新してから1msが経過したか否か判定する。現在のタイム102の値が、現在時刻レジスタの値を前回、更新したときの値から1msに対応する値以上大きくなっていった場合、判定はYESとなり、ステップ402に移行してそのレジスタの値のインクリメントを行った後、ステップ403に移行する。そうでない場合には、判定はNOとなってステップ406に移行する。

20

【0090】

ステップ403では、MIDI IN端子114がMIDIデータを受信したか否か判定する。ユーザが鍵盤装置130を操作した場合、鍵盤装置130はMIDIデータを出力することから、判定はYESとなってステップ404に移行する。そうでない場合には、即ちユーザが鍵盤装置130の操作を行っていないような場合には、判定はNOとなってステップ406に移行する。

【0091】

ステップ404では、MIDI IN端子114が受信したMIDIデータを送信器108に送って送信させる。続くステップ405では、現在時刻レジスタの現在値を送信器108に送り、それを時間データとして送信させる。その後、ステップ406に移行する。なお、上述したように、MIDIデータ、及び時間データは一つの packets にまとめて送信される。

30

【0092】

ステップ406～411では、音楽情報として音声データ（波形データ）を送信するための処理が行われる。

先ず、ステップ406では、音声情報として音声データを送信することを示す定数（0，0）を送信器108に送信させる。以降、同様に、波形バッファの容量を示すバイト数である256を音声データのバイト数として送信させ、続いてシリアル番号として変数SCの値を送信させる（ステップ407、408）。変数SCの値を送信させると、ステップ409でその値のインクリメントを行った後、ステップ410に移行する。

40

【0093】

ステップ410では、変数SHBFの値が指定する波形バッファに格納された波形データを送信器108に送って送信させる。その後は、ステップ411に移行して、変数BCの値の更新を行う。その更新は、音声データとして256バイトを送信することから、それまでの値に256を加算して得られる値を新たに変数BCに代入することで行う。なお、識別子である定数（0，0）から音声データは、実際には一つの packets にまとめて送信される。

50

## 【 0 0 9 4 】

ステップ 4 1 1 に続くステップ 4 1 2 では、ステップ 4 1 0 で送信器 1 0 8 に依頼した音声データの送信が終了するのを待つ。その送信が終了するのを待って、ステップ 4 1 3 に移行する。

## 【 0 0 9 5 】

音声データを送信することにより、音楽情報としての音声データの送信が完了する。このことから、ステップ 4 1 3 以降では、送信した音声データの再生を指示する再生指示コマンドを音楽情報として送信するための処理が行われる。

## 【 0 0 9 6 】

先ず、ステップ 4 1 3 では、音楽情報として再生指示コマンドを送信することを示す定数 ( 1 , 0 ) を送信器 1 0 8 に送信させる。以降は、変数 B C の値を音声データの総バイト数として送信させ、続けて録音時刻レジスタの値を時間データとして送信させる ( ステップ 4 1 4 、 4 1 5 ) 。 そのようにして再生指示コマンドを音楽情報として送信した後、一連の処理を終了する。なお、再生指示コマンドを兼ねる定数 ( 1 , 0 ) から時間データの送信は、実際には一つの packets にまとめて行われる。

10

## 【 0 0 9 7 】

このように、本実施の形態では、画像データ ( 楽譜のビットマップデータ ) は実際の演奏に係わる音楽情報の送信に先だって送信し、 M I D I データはそれを受信する度に送信し、ポイントデータは必要に応じて送信し、音声データは波形バッファの空容量がなくなる度に送信し、再生指示コマンドは音声データを送信する度に送信するようにしている。

20

## 【 0 0 9 8 】

なお、本実施の形態では、便宜的に音声データと再生指示コマンドを別にして送信するようにしているが、その再生指示コマンドは送信しなくても良い。音声データに、時間データ、及びそれを再生すべきか否かを判定するための各種データを付加して送信するようにしても良い。

## 【 0 0 9 9 】

次に、音楽情報を受信した場合の C P U 1 0 1 の動作について、図 5 を参照して詳細に説明する。その図 5 は、音楽情報受信処理の動作フローチャートである。

## 【 0 1 0 0 】

自分宛にデータが何時、送信されるかを予め知ることはできない。このことから、音楽情報受信処理は、受信器 1 0 7 がネットワーク ( ここでは L A N を想定 ) に接続されている場合に、電源がオンされた後、 C P U 1 0 1 が R O M 1 0 3 に格納されているプログラムを実行することで実現される。

30

## 【 0 1 0 1 】

なお、上述したようにして送信された音楽情報は、サーバに保持されることから、音楽情報の受信はサーバからそれをダウンロードすることで行われる。自分宛に送信されたデータの有無は、サーバに問い合わせることで確認する。

## 【 0 1 0 2 】

先ず、ステップ 5 0 1 では、音楽情報の受信開始に備えて、各種変数に初期値を代入することを含むイニシャライズを行う。各種変数への初期値の代入としては、送られてきた音声データのバイト数を保持するための変数 J B C 、音声データのシリアル番号を管理するための変数 J S C 、及び、送られてきた音声データの総バイト数を管理するための変数 H S S R には各々 0 を代入し、受信した音声データを格納するための受信波形バッファを確保し、受信した音声データの格納先を管理するための変数 J H B F には受信波形バッファ A を示す A を代入する。そのようなことを行ってイニシャライズが終了すると、次にステップ 5 0 2 に移行する。

40

## 【 0 1 0 3 】

ステップ 5 0 2 では、受信器 1 0 7 により受信した音楽情報の内容を判別してそれに応じた処理を行うラベル 3 処理を実行する。続くステップ 5 0 3 では、サーバとの接続が断られたか否かを判定する。サーバとの接続を切断する操作をユーザが行ったような場合、音楽

50

情報の受信は終了したとして判定はYESとなり、一連の処理を終了する。そうでない場合には、判定はNOとなってステップ502に戻り、ラベル3処理を実行する。それにより、音楽情報の受信が終了するまでの間、ステップ502のラベル3処理を繰り返し実行する。

#### 【0104】

図6は、上記ステップ502として実行されるラベル3処理の動作フローチャートである。次に、図6を参照して、ラベル3処理について詳細に説明する。

送信元では、音楽情報の種類によって時間データを付加して送信する。その時間データは、送信元が1ms単位で単に計時した時刻である。このことから、本実施の形態では、音楽情報の処理用の変数を用意し、その変数を用いて時刻を計時し、そのようにして計時した時刻を時間データとして受信した時刻と比較することにより、時間データで指定されるタイミングで音楽情報を処理するようにしている。なお、その変数については、以降、便宜的に発音時刻レジスタと呼ぶことにする。

10

#### 【0105】

まず、ステップ601では、上記発音時刻レジスタの値を前回、更新してから1ms経過したか否か判定する。タイマ102の現在値が、そのレジスタの値を前回、更新したときの値から1msに対応する値以上大きかった場合、判定はYESとなり、ステップ602でその値のインクリメントを行った後、ステップ603に移行する。そうでない場合には、判定はNOとなってステップ603に移行する。

#### 【0106】

本実施の形態では、音楽情報として受信したMIDIデータを格納する領域（以降、音楽データバッファと呼ぶ）をRAM104に確保し、MIDIデータを受信する度に、そのバッファにMIDIデータを時間データとともに格納して、時間データで指定されるタイミングでMIDIデータを処理するようにしている。ステップ603では、その音楽データバッファに格納された未処理のMIDIデータのなかで処理タイミングとなったものがあるか否か判定する。時間データとして付加された時刻が発音時刻レジスタの値で表される時刻よりも早いMIDIデータが音楽データバッファに格納されていた場合、判定はYESとなり、ステップ604で該当するMIDIデータを処理した後、ステップ605に移行する。そうでない場合には、判定はNOとなってステップ606に移行する。なお、MIDIデータの処理は、CPU101が、RAM104のその音楽データバッファから該当するMIDIデータを読み出し、それをMIDI OUT端子113を介して音源装置120に送出することで行われる。

20

30

#### 【0107】

ステップ605では、受信器107を介してサーバに問い合わせを行い、自分宛に送信されたデータがあればダウンロードさせて受け取り、そのデータ中の先頭の2ワード分を取り出す。その2ワードのデータは、音楽情報がMIDIデータであればそのMIDIデータと時間データであり、それ以外の音楽情報では2バイトの識別子である。ステップ605に続くステップ606以降の処理は、そのデータから特定される音楽情報の種類に応じて行われる。

#### 【0108】

まず、ステップ606では、そのデータで表現された値が(0,0)か否か判定する。音楽情報として音声データを受信器107が受信（サーバからダウンロード）した場合、判定はYESとなり、ステップ607で音声データの受信に対応するためのラベル4処理を実行した後、一連の処理を終了する。そうでない場合には、即ち音声データ以外の音楽情報を受信した場合には、判定はNOとなってステップ608に移行する。

40

#### 【0109】

ステップ608では、2ワード分のデータで表現された値が(1,0)か否か判定する。音楽情報として再生指示コマンドを受信器107が受信した場合、判定はYESとなり、ステップ609で再生指示コマンドの受信に対応するためのラベル5処理を実行した後、一連の処理を終了する。そうでない場合には、即ち音声データ、及び再生指示コマンド以

50

外の音楽情報を受信した場合には、判定はNOとなってステップ610に移行する。

【0110】

ステップ610では、2ワード分のデータで表現された値が(2, 0)か否か判定する。音楽情報として画像データを受信器107が受信した場合、判定はYESとなり、ステップ611で画像データの受信に対応するためのラベル6処理を実行した後、一連の処理を終了する。そうでない場合には、即ち音声データ、再生指示コマンド、及び画像データ以外の音楽情報を受信した場合には、判定はNOとなってステップ612に移行する。

【0111】

ステップ612では、2ワード分のデータで表現された値が(3, 0)か否か判定する。音楽情報としてポイントデータを受信器107が受信した場合、判定はYESとなり、ステップ613で再生指示コマンドの受信に対応するためのラベル7処理を実行した後、一連の処理を終了する。そうでない場合には、即ち音楽情報としてMIDIデータを受信した場合には、判定はNOとなってステップ614に移行する。

【0112】

ステップ614では、音楽情報として受信したMIDIデータを時間データとともにRAM104の音楽データバッファに格納する。その格納を行った後、一連の処理を終了する。

【0113】

このようにして、ラベル3処理では、時間データで指定されたタイミングとなったか否か確認しながら、音楽情報の処理、その受信に対処するための処理を行うようになっている。以降は、そのラベル3処理内で実行される各種サブルーチン処理について詳細に説明する。

【0114】

図7は、上記ステップ607として実行されるラベル4処理の動作フローチャートである。次に、図7を参照して、ラベル4処理について詳細に説明する。そのラベル4処理は、音声データであることを示す識別子、音声データのバイト数、シリアル番号、音声データの順に並べられた音楽情報の受信に対応するための処理である。

【0115】

まず、ステップ701では、受信器107から受け取った受信データ中から、識別子である定数の次に位置するバイト数を取り出し、続くステップ702でそれを変数JBCに代入する。それが終了すると、ステップ703で受信データ中からシリアル番号を取り出した後、ステップ704に移行して、受信したシリアル番号が音声データの受信回数のカウント用に用意した変数JSCの値と等しいか否か判定する。送信元から送信された音声データを全て正常に受信していた場合、それらの値が一致することから判定はYESとなってステップ705に移行する。そうでない場合には、判定はNOとなってステップ709に移行する。

【0116】

ステップ705では、シリアル番号の次に位置する256バイトの音声データを受信データ中から取り出す。続くステップ706では、それを変数JHBFの値で指定される受信波形バッファに格納する。それが終了すると、ステップ707で変数JSCの値をインクリメントし、次にステップ708で変数HSSRの値を更新した後、一連の処理を終了する。

【0117】

上記ステップ704でのNOの判定は、送信元から送信された音声データのなかで正常に受信できなかったものが存在していることを意味する。音声データの受信にエラーが発生していることを意味する。このことから、ステップ709では、そのエラーに対応するためのラベル8処理を実行する。そのラベル8処理を実行した後、一連の処理を終了する。

【0118】

図8は、上記ラベル8処理の動作フローチャートである。音声データの受信にエラーが発生した場合、そのエラーによって異常な音を発音させてし

10

20

30

40

50

もう可能性が生じる。このことから、ラベル 8 処理では、受信した音声データを破棄する処理をステップ 8 0 1 で行う。それを行った後、一連の処理を終了する。

【 0 1 1 9 】

図 9 は、上記音楽情報受信処理でステップ 6 0 9 として実行されるラベル 5 処理の動作フローチャートである。次に、図 9 を参照して、その処理について詳細に説明する。

【 0 1 2 0 】

ラベル 5 処理は、再生指示コマンドの受信に対応するための処理である。送信元は、音その再生指示コマンドを示す識別子、それまでに送信した音声データの総バイト数、時間データの順に並べて音楽情報として送信する。なお、その総バイト数は、変数 B C の値である。

10

【 0 1 2 1 】

まず、ステップ 9 0 1 では、受信器 1 0 7 から受け取った受信データ中から、識別子である定数の次に位置する総バイト数を取り出す。続くステップ 9 0 2 では、その総バイト数に変数 H S S R の値と等しいか否か判定する。再生指示コマンドを送信したときの送信元における変数 B C の値が変数 H S S R の値と一致していた場合、判定は Y E S となってステップ 9 0 3 に移行する。そうでない場合には、判定は N O となってステップ 9 0 7 に移行する。

【 0 1 2 2 】

ステップ 9 0 3 では、総バイト数の次に位置する時間データを受信データ中から取り出して、それを発音時刻レジスタに代入する。その後、ステップ 9 0 4 に移行する。

20

【 0 1 2 3 】

再生指示コマンドに先だって送信される音声データは、マイク 1 1 1 が集音した音のデータであり、A D コンバータ 1 1 2 から随時 C P U 1 0 1 に送られる。このことから、普通は受信器 1 0 7 が最初に受信する種類の音楽情報となる。その音声データは他のデータと比較してバイト数が多いため、送信時間が最も変動すると予想される。このようなことから、本実施の形態では再生指示コマンドに付加された時間データを発音時刻レジスタに代入している。それにより、受信側での音楽情報の受信による演奏は、音声データの再生を基準として行うようにしている。再生指示コマンドに付加された時間データを発音時刻レジスタに代入することにより、M I D I データを処理するタイミングは、図 6 のラベル 3 処理のステップ 6 0 3 で説明したように、それに付加された時間データを発音時刻レジスタの値と単に比較することで特定できることになる。

30

【 0 1 2 4 】

ステップ 9 0 4 では、再生する対象となる音声データが格納された受信波形バッファを管理するための変数 O H B F に変数 J H B F の値を代入する。即ち直前に受信された音声データの格納先の受信波形バッファを示す値を変数 O H B F に代入する。続くステップ 9 0 5 では、変数 J H B F の値の反転、即ちそれまでの値が A であれば B に、或いは、それまでの値が B であれば A にする書き換えを行う。そうした後、ステップ 9 0 6 に移行して、変数 O H B F の値で指定される受信波形バッファに格納された音声データの再生を開始する。その再生を開始した後一連の処理を終了する。

【 0 1 2 5 】

なお、特に詳細な説明は省略するが、音声データの再生は、その受信波形バッファに格納された音声データ中の 1 サンプル分のデータを、サンプリング周期に合わせて順次読み出し、それを D A コンバータ 1 0 9 に出力することで行っている。

40

【 0 1 2 6 】

ステップ 9 0 2 での N O の判定は、上述したように、音声データの受信に何らかのエラーが発生した可能性があることを意味する。このことから、ステップ 9 0 7 では、そのエラーに対応するためのラベル 9 処理を実行して、音声データの再生を中止させる。そのラベル 9 処理を実行した後、一連の処理を終了する。

【 0 1 2 7 】

図 1 0 は、そのラベル 9 処理の動作フローチャートである。

50

このラベル 9 処理では、図 10 に示すように、ステップ 1001 で受信データ中の時間データを取り出すと、次に変数 JBC、及び HSSR の値をクリア、即ち 0 を代入（ステップ 1002、1003）した後、一連の処理を終了する。変数 HSSR の値をクリアすることにより、それ以降では、上記ステップ 902 の判定は常に NO となる。このため、正常に受信することができた音声データの再生の終了とともに、音声データの再生が行われなくなる。それにより、エラーの発生によって異常な音を発音させるようなことが回避される。

#### 【0128】

図 11 は、図 6 のラベル 3 処理でステップ 611 として実行されるラベル 6 処理の動作フローチャートである。

画像データ（ここでは、楽譜のビットマップデータが前提）の受信に対応するためのラベル 6 処理では、識別子（2, 0）の次に位置する画像データを受信データ中から取り出し、それを液晶表示器 106 が備えた V RAM に転送する処理をステップ 1101 で行う。それにより、液晶表示器 106 に楽譜の表示が行えるようにした後、一連の処理を終了する。

#### 【0129】

図 12 は、図 6 のラベル 3 処理でステップ 613 として実行されるラベル 7 処理の動作フローチャートである。

ポイントデータ（ここでは、楽譜上の演奏位置を示すカーソルの表示位置を指定するデータである）の受信に対応するためのラベル 7 処理では、識別子（3, 0）の次に位置する X 座標データを受信データ中から取り出してカーソルポインタ 105 のカーソル X レジスタに格納した（ステップ 1201、1202）後、同様に、Y 座標データを受信データ中から取り出してカーソル Y レジスタに格納する（ステップ 1203、1204）ことを行う。そのようにしてカーソルの表示位置の変更を指示した後、一連の処理を終了する。

#### 【0130】

なお、第 1 の実施の形態では、イベントデータとして MIDI データ、音声データ、再生指示コマンド、画像データ、及びポイントデータの計 5 種類を送受信することを想定しているが、当然のことながら、音楽情報として送信する内容は、それに限定するものではない。それ以上であっても、それ以下であっても良い。また、それらを送信する際の形式も任意に変更して良い。

#### 【0131】

上記音声データについては、本実施の形態では常に送信するようにしているが、必ずしもそうしなくて良い。例えば音量レベルを設定しておき、その音量レベルよりも大きな音量の音をマイク 111 が拾っているときだけ音声データを送信するようにしても良い。音声データを圧縮して送信するようにしても良い。そのようにして受信側で再生すべき価値のある音声データだけを送信したり、或いはそのデータ量を低減させた場合には、それ以外の種類の音楽情報をより大量に送信することができる。音声データの受信にエラーが発生した場合の処置については、直ちにミュートさせるのではなく、クロスフェードさせてエラーの発生を目立たなくさせても良い。

#### 【0132】

また、第 1 の実施の形態では、音楽情報の送受信を行えるようになっているが、音楽情報の送信だけ、或いはその受信だけを行えるようにしても良い。その送受信では、処理（送信、或いは受信）の対象とする音楽情報の種類を任意に選択できるようにすることが望ましい。

< 第 2 の実施の形態 >

近年、自動演奏（ここでは自動伴奏を含む）を行う自動演奏機能は、その専用の装置であるシーケンサーを始めとして、様々な装置に搭載されている。ユーザは、その機能を用いることにより、合奏を楽しむことができる。

#### 【0133】

自動演奏は、それを行うための演奏データ（シーケンスデータ）を処理することで行われ

10

20

30

40

50

る。その演奏データは、普通、演奏上のイベントの内容を表すイベントデータ（例えばMIDIデータ）とそれを処理すべきタイミングを表す時間データとからなるデータ対を1つの処理単位として、それを処理する順序に並べた形で構成される。そのような構成であることから、或るパートを演奏させないというようなことはできるが、別のパートを付加して演奏を行うといったようなことはできない。第2の実施の形態は、そのことに着目し、自動演奏として行える内容を付加できるようにしたものである。

**【0134】**

第2の実施の形態は、上記第1の実施の形態による音楽情報受信装置に、自動演奏の内容を付加できる機能（音楽情報編集装置）を搭載させることで実現させたものである。その構成は、基本的に第1の実施の形態のそれ（図1参照）と同じである。このことから、第2の実施の形態の説明は、第1の実施の形態の説明で用いた符号をそのまま流用して、図1を参照しつつ行うことにする。

10

**【0135】**

図13は、第2の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の利用方法を説明する図である。先ず、図13を参照して、その利用方法、及びそれを可能にさせる動作について説明する。

**【0136】**

図13中の1301～1304は、例えば、それぞれが図1に示す構成を有する楽音発生装置である。矢印は、音楽情報がどこからどこに送信されるのかを示している。即ち、楽音発生装置1301に、各楽音発生装置1302～1304が音楽情報を送信することを示している。

20

**【0137】**

各楽音発生装置1302～1304は、そのユーザがスイッチ群115を操作して指定した送信先に、同じくスイッチ群115を操作して選択した演奏データを送信する。その送信は、CPU101が、選択された演奏データをROM103から読み出し、それを送信器108に送ることで行われる。

**【0138】**

それらの装置1302～1304から送信された演奏データは、それらとは別の楽音発生装置1301に受信される。その装置1301のCPU101は、受信した演奏データを格納する領域（以降、便宜的に音楽データバッファと呼ぶ）をRAM104に前もって確保し、そのバッファに、受信器107が受信した演奏データを格納する。本実施の形態では、演奏データの受信は各装置1302～1304毎にハンドシェイクで行うようにしている。

30

**【0139】**

各装置1302～1304からの演奏データの受信が終了すると、装置1301のCPU101は、イベントデータ（ここではMIDIデータである）に付加された時間データを基に、その時間データで指定される処理タイミングが早い順にソートする。イベントデータと時間データからなるデータ対の処理順序に従った並べ換えを行う。その並べ換えを行った後、それを行った演奏データの再生を開始する。

**【0140】**

上記時間データによるタイミングの管理方法は、或る基準となる時間的位置から経過した時間で表す絶対時間系の管理方法と、直前のイベントが発生してから経過した時間で表す相対時間系の管理方法と、の2種類に大別される。本実施の形態で採用しているのは前者である。このため、上記ソートに応じた時間データへの操作は行っていない。

40

**【0141】**

上記のように、時間データを基にソートを行うことで、各演奏データを構成するイベントデータの処理タイミングを維持させつつ、複数の演奏データを適切な形で一つにまとめることができる。このため、ユーザは、様々な演奏データを任意に組み合わせるとともに、それらの演奏データを各々、適切に再生させられるようになる。それにより、ユーザが行える音楽表現の幅が広がるとともに、より多様な楽しみ方ができるようになることから、

50

ユーザにとっての利便性が向上することになる。それ以外には、演奏データを分割して保存することができることから、その全てを盗難されたり、その全てを削除するといったことが一度に行えなくなる。そのため、安全面上の効果も得ることができる。

【0142】

なお、本実施の形態では、組み合わせる演奏データを他の装置から受信するようにしているが、別の方法でその演奏データを取得するようにして良い。フロッピーディスクやCD-ROM、或いはICカード等の記録媒体にアクセスして演奏データを取得しても良い。当然のことながら、ROM103、或いは記録媒体に格納された演奏データを複数選択させ、選択された演奏データを組み合わせられるようにしても良い。

【0143】

次に、上述した制御を行うCPU101の動作について、図14～図19に示す各種動作フローチャートを参照して詳細に説明する。ここでは、理解を容易にするために、音楽情報の送信、及びその受信と編集に着目して説明することにする。

【0144】

図14は、音楽情報送信処理の動作フローチャートである。音楽情報（ここでは演奏データ）の送信に着目して、その処理の流れを抜粋して示したものである。始めに、図14、及び図15を参照して、演奏データを送信する際のCPU101の動作について詳細に説明する。

【0145】

なお、図14に示す音楽情報送信処理（図15のそのサブルーチン処理を含む）は、ユーザがスイッチ群115を操作して、送信させる演奏データを指定した場合に、CPU101がROM103に格納されているプログラムを実行することで実現される。

【0146】

先ず、ステップ1401では、RAM104に音楽データバッファを確保するといったイニシャライズを行う。続くステップ1402では、その音楽データバッファに、ユーザが指定した演奏データをROM103から読み出して格納する。その格納が終了すると、ステップ1403に移行する。

【0147】

本実施の形態では、編集に用いられることを前提とする演奏データは、その送信を要求してきた相手先に、その要求を待っている。このことから、ステップ1403では、その要求が送信されてきたか否か判定する。送信先がその要求を送信した場合、判定はYESとなり、ステップ1404で演奏データを送信するラベル1処理を実行した後、ステップ1405に移行する。そうでない場合には、判定はNOとなってステップ1405に移行する。

【0148】

ステップ1405では、演奏データの送信が終了したか否か判定する。送信すべき演奏データを送信先に全て送信した、或いはその送信の中止をユーザがスイッチ群115を介して指示したような場合、判定はYESとなって一連の処理を終了する。そうでない場合には、判定はNOとなってステップ1403に戻り、演奏データの送信を継続させる必要があるか否か判定する。それにより、送信を要求してきた全ての送信先に、送信すべき演奏データを全て送信するようにしている。

【0149】

図15は、上記ステップ1404として実行されるラベル1処理の動作フローチャートである。

このラベル1処理では、先ず、ステップ1501で楽音データバッファに格納した演奏データの送信を行う。続くステップ1502では、そのバッファ内に送信すべきデータが無いか否か判定する。全てのデータを送信した場合、判定はYESとなり、全てのデータを送信した旨を示すメッセージを送信先に送信した後、一連の処理を終了する。そうでない場合には、判定はNOとなってステップ1501に戻り、未送信分のデータの送信を行う。

10

20

30

40

50

## 【 0 1 5 0 】

上記のように、ラベル 1 処理では、ステップ 1 5 0 1 と 1 5 0 2 で処理ループを形成しており、未送信分のデータが存在している間、その処理ループが繰り返し実行される。それにより、演奏データを、予め定めたデータ量に分けて送信するようにしている。なお、その送信は、CPU 1 0 1 が音楽データバッファから読み出したデータにヘッダを付加して送信器 1 0 8 に送ることで行われる。

## 【 0 1 5 1 】

図 1 6 は、音楽情報受信処理の動作フローチャートである。次に、この図 1 6、更には図 1 7 ~ 図 1 9 を参照して、演奏データの受信に係わる CPU 1 0 1 の制御動作について詳細に説明する。

10

## 【 0 1 5 2 】

演奏データは、ユーザがスイッチ群 1 1 5 を介して指定した送信元から受信するようになっている。音楽情報受信処理は、その送信元を指定した後、スイッチ群 1 1 5 を介して演奏データの受信を指示した場合に、CPU 1 0 1 が ROM 1 0 3 に格納されているプログラムを実行することで実現される処理である。

## 【 0 1 5 3 】

先ず、ステップ 1 6 0 1 では、演奏データの受信開始に備えてイニシャライズを行う。それにより、演奏データの送信元を管理するための変数 SAR の値はクリア、即ち 0 を代入し、受信した演奏データを格納する領域を RAM 1 0 4 に確保する。それが終了すると、ステップ 1 6 0 2 に移行する。なお、上記領域については、以降、便宜的に受信音楽データバッファと呼ぶことにする。

20

## 【 0 1 5 4 】

ステップ 1 6 0 2 では、送信元に演奏データの送信を要求してその演奏データを受信するラベル 2 処理を実行する。続くステップ 1 6 0 3 では、演奏データに対する全ての処理が終了したか否か判定する。受信した演奏データは、編集を行った後、再生するようにしている。このため、その再生が終了した、或いはその途中でユーザがスイッチ群 1 1 5 を介して終了を指示した場合、判定は YES となって一連の処理を終了する。そうでない場合には、判定は NO となり、ステップ 1 6 0 2 に戻ってラベル 2 処理を実行する。

## 【 0 1 5 5 】

図 1 7 は、上記ステップ 1 6 0 2 として実行されるラベル 2 処理の動作フローチャートである。次に、図 1 7 を参照して、ラベル 2 処理について詳細に説明する。

30

## 【 0 1 5 6 】

演奏データの送信元は複数、指定することができるようになっている。CPU 1 0 1 は、ユーザがスイッチ群 1 1 5 により送信元として入力したデータ（ここではアドレス）を RAM 1 0 4 に格納して参照することにより、送信元を順次、切り替えながら演奏データを受信する。変数 SAR は、そのような演奏データの送信元の切り換えを実現するために用いられる。

## 【 0 1 5 7 】

先ず、ステップ 1 7 0 1 では、変数 SAR の値で指定される送信元に演奏データの送信を要求する。続くステップ 1 7 0 2 では、受信器 1 0 7 を介してサーバに問い合わせを行い、演奏データが送信されたか否か判定する。送信元が演奏データを送信した場合、判定は YES となってステップ 1 7 0 3 に移行する。そうでない場合には、判定は NO となってステップ 1 7 0 1 に戻る。それにより、演奏データが送信されるのを待つ。

40

## 【 0 1 5 8 】

ステップ 1 7 0 3 では、受信器 1 0 7 によりサーバから演奏データを受信して、その演奏データと同一の演奏データが受信音楽データバッファに格納されていないか否か判定する。受信された演奏データ中の MIDI データ、及びそれに付加された時間データの内容が受信音楽データバッファに格納されている演奏データ中のそれらとっていた場合、判定は YES となり、ステップ 1 7 0 4 で受信された演奏データを格納した後、ステップ 1 7 0 5 に移行する。そうでない場合には、判定は NO となってステップ 1 7 0 1 に戻り、演奏

50

データの送信を要求する。

【0159】

ステップ1705では、受信すべきデータがないか否か判定する。受信したデータ中に演奏データの送信が終了した旨を示す識別子が挿入されていた場合、判定はYESとなり、ステップ1706で変数SARの値をインクリメントした後、ステップ1707に移行する。そうでない場合には、判定はNOとなってステップ1701に戻る。

【0160】

ステップ1707では、演奏データを受信していない送信元がないか否か判定する。変数SARの値に対応する送信元が存在していない場合、判定はYESとなってステップ1708に移行し、そうでない場合には、判定はNOとなってステップ1701に戻る。

10

【0161】

ステップ1707でのYESの判定は、ユーザが指定した全ての送信元からの演奏データの受信が終了したことを意味する。このことから、ステップ1708では、受信した演奏データを編集して再生するラベル3処理を実行する。そのラベル3処理を実行した後、一連の処理を終了する。

【0162】

図18は、上記ステップ1708として実行されるラベル3処理の動作フローチャートである。次に、図18を参照して、ラベル3処理について詳細に説明する。

【0163】

ラベル3処理の実行時には、受信音楽データバッファに、受信された演奏データを送信元別に格納している。ラベル3処理は、それらの演奏データを対象に行われる。

20

【0164】

まず、ステップ1801では、MIDIデータに付加された時間データを参照することにより、各演奏データ中からMIDIデータを処理すべきタイミングが早い順に取り出して並べるソートを行う。それにより、複数の演奏データを、MIDIデータの処理タイミングを維持させつつ一つにまとめる。

【0165】

ステップ1801に続くステップ1802では、演奏時刻レジスタに、時間データのなかで最も指定するタイミングが早い時間データを代入する。言い換えれば、最初に処理すべきMIDIデータに付加されている時間データを演奏時刻レジスタに代入する。なお、上記演奏時刻レジスタは、MIDIデータを時間データに従って処理するために用いる変数であり、例えばタイマ102の値を参照して、1msが経過する毎にその値をインクリメントするようになっている。

30

【0166】

上記ステップ1802は、演奏データを再生を開始するうえでのイニシャライズとして行われる。それに続くステップ1803では、演奏データを再生するラベル4処理を実行する。ラベル3処理を構成する一連の処理は、そのラベル4処理を実行した後に終了する。

【0167】

図19は、上記ステップ1803として実行されるラベル4処理の動作フローチャートである。次に、その処理について詳細に説明する。このラベル4処理は、編集後の演奏データを対象にして行われる。

40

【0168】

まず、ステップ1901では、演奏時刻レジスタの値の更新を行ってから1msが経過したか否か判定する。その更新を行ったときのタイマ102の値とそれの現在値に1msに対応する値以上の差があった場合、判定はYESとなり、ステップ1902でその値をインクリメントした後、ステップ1903に移行する。そうでない場合には、判定はNOとなってステップ1903に移行する。

【0169】

ステップ1903では、演奏データ中で次に処理すべきMIDIデータの処理タイミングがきたか否か判定する。そのMIDIデータに付加された時間データの値が演奏時刻レジ

50

スタの値以下であった場合、判定はYESとなり、ステップ1904でそのMIDIデータを処理した後、ステップ1905に移行する。そうでない場合には、判定はNOとなってステップ1901に戻る。なお、MIDIデータの処理は、CPU101が、それをMIDI OUT端子113に送出することで行われる。

【0170】

ステップ1905では、処理の対象としている演奏データ中に、未処理のMIDIデータがあるか否か判定する。全てのMIDIデータを処理していない場合、判定はYESとなり、ステップ1906で次に処理すべきMIDIデータに処理対象を変更した後、ステップ1901に戻る。そうでない場合には、即ち演奏データの再生が終了した場合には、判定はNOとなって一連の処理を終了する。なお、上記処理対象の変更は、例えば処理対象とするMIDIデータを管理するための変数に、次に処理すべきMIDIデータが格納されているアドレスの値を代入することで行われる。

10

【0171】

このように、ラベル4処理は、演奏時刻レジスタの値を随時、更新しつつ、時間データで指定されたタイミングでMIDIデータを処理していくようになっている。

【0172】

なお、第2の実施の形態では、複数の演奏データを一つにまとめる編集を行うようにしているが、その編集を行う替わりに、例えばMIDIデータを処理する毎に複数の演奏データのなかから次に処理すべきMIDIデータを探しだすようにしても良い。しかし、相対時間系の管理方法を採用している場合には、そのようにするとタイミングを特定するために面倒な操作を演奏データの再生と並行に行わなければならないことから、編集を行った後に演奏データを再生することが望ましい。

20

【0173】

また、第2の実施の形態では、MIDIデータで構成された演奏データを編集の対象としているが、当然のことながら、その演奏データはMIDIデータ以外のデータ、例えば音声データを用いて構成されていても良い。その演奏データは送信元に予め格納されたものであるが、演奏を開始した時点を中心とした時間データを付加するといったように、各送信元が計時の基準とする時点を予め決めておいた場合には、随時、生成して送信される音楽情報を編集の対象とすることもできる。

【0174】

また、送信させる演奏データについては、各装置1302～1304のユーザが指定するようにしているが、その演奏データの送信を要求する側が演奏データを指定できるようにしても良い。

30

【0175】

更に、本実施の形態では、送信されてきた演奏データを対象にその編集を行うようにしているが、同じ記録媒体に格納されている演奏データを対象に編集を行うようにしても良い。このことから明らかなように、音楽情報編集装置は、音楽情報を受信できる装置だけでなく、様々な装置に搭載することができる。

<第3の実施の形態>

MIDIの普及、即ち音楽情報を容易に送受信することができるようになったことは、装置を特定の用途で使うといったように、装置の用途の選択範囲を広げている。このことから、複数の装置を組み合わせるユーザが自分に合ったシステムを構築し、そのシステムを通して音楽を楽しむことも広く行われている。第3の実施の形態は、音楽情報の送受信によって創造される音楽の可能性に着目して、用途の選択範囲をより広げられるようにしたものである。

40

【0176】

第3の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の構成は、上記第1の実施の形態におけるそれ(図1参照)と基本的に同じである。このことから、第3の実施の形態の説明は、第1の実施の形態の説明で用いた符号を基本的にはそのまま流用して、図1を参照しつつ行うことにする。

50

## 【 0 1 7 7 】

図 2 0 は、第 3 の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の利用方法を説明する図である。先ず、図 2 0 を参照して、その利用方法、及びそれを可能にさせる動作について説明する。

## 【 0 1 7 8 】

図 2 0 中の 2 0 0 1 ~ 2 0 0 5 は、例えば、それぞれが図 1 に示す構成を有する楽音発生装置である。矢印は、音楽情報がどこからどこに送信されるのかを示している。即ち、楽音発生装置 2 0 0 1 は楽音発生装置 2 0 0 2 と音楽情報の送受信を行い、その楽音発生装置 2 0 0 2 は、楽音発生装置 2 0 0 3 ~ 2 0 0 5 に音楽情報を送信することを示している。第 1 及び第 2 の実施の形態と同様に、それらの装置 2 0 0 1 ~ 2 0 0 5 は LAN を介して接続されている。

10

## 【 0 1 7 9 】

図 2 0 に示すシステムにおいて、上記楽音発生装置 2 0 0 1 は、演奏に用いられることを前提とし、楽音発生装置 2 0 0 2 は、各装置 2 0 0 1、及び 2 0 0 3 ~ 2 0 0 5 に音楽情報を提供することを前提とし、楽音発生装置 2 0 0 3 ~ 2 0 0 5 は、受信した音楽情報を再生することを前提として用いられる。このことから、以降、便宜的に、楽音発生装置 2 0 0 1 は演奏装置、楽音発生装置 2 0 0 2 はデータベース装置、楽音発生装置 2 0 0 3 ~ 2 0 0 5 は受信装置と呼ぶことにする。

## 【 0 1 8 0 】

図 2 0 に示すシステムを構成する各装置 2 0 0 1 ~ 2 0 0 5 は、以下のように動作する。各装置 2 0 0 1 ~ 2 0 0 5 が前提とする動作は、例えばユーザがスイッチ群 1 1 5 を操作してそれに対応するモードを設定することで行われる。各装置 2 0 0 1 ~ 2 0 0 5 間の接続は、例えばデータベース装置として動作することを指定するモードが設定された装置 2 0 0 2 において、演奏装置 2 0 0 1 と受信装置 2 0 0 3 ~ 2 0 0 5 とで分けて相手先をスイッチ群 1 1 5 により指定することで行われる。データベース装置 2 0 0 2 は、各装置 2 0 0 1、及び 2 0 0 3 ~ 2 0 0 5 に、ユーザがスイッチ群 1 1 5 を介して指定した演奏データを ROM 1 0 3 から読み出し、それを以下のように音楽情報として提供する。

20

## 【 0 1 8 1 】

演奏装置 2 0 0 1 に対しては、その演奏データを構成する M I D I データに付加された時間データを参照し、その時間データで指定されるタイミングで M I D I データを時間データとともに随時送信する。

30

## 【 0 1 8 2 】

そのようにして演奏データが送信されてくる演奏装置 2 0 0 1 は、時間データに基づいて M I D I データを処理する。それにより、データベース装置 2 0 0 2 から送信された演奏データを再生する。データベース装置 2 0 0 2 へは、演奏データの再生に応じて行ったユーザの合奏内容を送信する。具体的には、鍵盤装置 1 3 0 に対して行った操作内容を示す M I D I データに時間データを付加して送信する。その時間データは、データベース装置 2 0 0 2 から送信されてきた時間データを基に生成したものである。

## 【 0 1 8 3 】

本実施の形態では、上述したように、データベース装置 2 0 0 2 にはユーザが行った演奏内容のみを送信している。これは、演奏装置 2 0 0 1 が受信した演奏データはデータベース装置 2 0 0 2 が持っているためである。言い換えれば、それを送信しなくても不具合が生じることをデータベース装置 2 0 0 2 側で回避することができるためである。そのように、データベース装置 2 0 0 2 から受信した演奏データの送信を回避することにより、データベース装置 2 0 0 2 と演奏装置 2 0 0 1 間で送受信するデータ量が全体的に見て低減することから、それらを接続する伝送路の使用効率を向上させることができる。

40

## 【 0 1 8 4 】

データベース装置 2 0 0 2 は、各受信装置 2 0 0 3 ~ 2 0 0 5 に対しては、演奏装置 2 0 0 1 に送信した演奏データに加えて、その演奏装置 2 0 0 1 から受信した演奏データを送信する。その送信は、演奏装置 2 0 0 1 との間のデータ送信に或る程度の時間がかかるこ

50

とから、その演奏装置 2001 から送信されてきた時間データを基に行っている。それにより、それらの演奏データを再生する際にタイミング的なズレが発生するのを回避させている。

【0185】

そのようにして演奏データが送信されてくる各受信装置 2003 ~ 2005 は、演奏装置 2001 と同様に、時間データに基づいて M I D I データを処理する。それにより、データベース装置 2002 自体が持っている演奏データの再生による演奏と、演奏装置 2001 の鍵盤装置 130 に対してユーザが行った演奏の合奏を再現する。

【0186】

このように、データベース装置 2002 は、各送信先に対して異なる内容の演奏データ（音楽情報）を送信できるようになっている。これは、各送信先（装置）毎に役割を異ならせたり、その役割の内容をより細かく指定するといったことが容易に行えることを意味する。それにより、例えば或る装置は演奏用、複数の装置をそれぞれ異なるパートの発音用といったように、各装置の用途を細かく設定することができる。このため、装置を使用するうえでの選択の幅が広がって利便性が向上し、ユーザはシステムを構成する装置をより所望する形で使用することができる。また、その利便性の向上によって、ユーザが行える音楽表現の幅を広げられることから、新しい音楽の創造にも貢献することができる。

10

【0187】

次に、上記各装置 2001 ~ 2005 の動作を実現させるそれに搭載された CPU 101 の制御動作について、図 21 ~ 図 26 を参照して詳細に説明する。ここでは、理解を容易にするために、演奏装置 2001、データベース装置 2002、受信装置 2003 ~ 2005 の各装置の機能を実現させる制御動作を、装置別に分けて説明することにする。

20

【0188】

図 21 は、演奏装置 2001 に搭載された CPU 101 が実行する音楽情報送受信処理の動作フローチャートである。始めに、図 21 を参照して、演奏装置 2001 に搭載された CPU 101 の動作について詳細に説明する。

【0189】

なお、その音楽情報送受信処理は、楽音発生装置が演奏装置として機能することをスイッチ群 115 によりユーザが指定した場合に、CPU 101 が ROM 103 に格納されたプログラムを実行することで実現される。

30

【0190】

先ず、ステップ 2101 では、イニシャライズを行う。それにより、各種変数にはその値のクリアを含めて予め定められた値を代入し、受信した演奏データを一時的に保持しておく領域（以降、受信音楽データバッファと呼ぶ）を RAM 104 に確保する。続くステップ 2102 では、データベース装置 2002 との間で音楽情報（ここでは、M I D I データ単位でやりとりする演奏データ）の送受信を行うためのラベル 1 処理を実行する。ステップ 2103 にはその後に移行する。

【0191】

ステップ 2103 では、音楽情報の送受信が終了したか否か判定する。例えばデータベース装置 2002 から予め定めた時間以上、演奏データが送信されなかった、或いはユーザがスイッチ群 115 を介して演奏データの送受信の中止を指示したような場合、判定は Y E S となり、ここで一連の処理を終了する。そうでない場合には、判定は N O となってステップ 2102 に戻り、ラベル 1 処理を再度実行する。

40

【0192】

上記ステップ 2102 とステップ 2103 とは処理ループを形成しており、そのステップ 2103 の判定が Y E S となるまで繰り返し実行される。それにより、データベース装置 2002 から M I D I データを順次受信して処理することによる自動演奏を行うとともに、ユーザが鍵盤装置 130 を操作した行った演奏内容を M I D I データの形でデータベース装置 2002 に送信する。

【0193】

50

図 2 2 は、上記ステップ 2 1 0 2 として実行されるラベル 1 処理の動作フローチャートである。次に、図 2 2 を参照して、ラベル 1 処理について詳細に説明する。

【 0 1 9 4 】

データベース装置 2 0 0 2 は、M I D I データに時間データを付加して送信する。その時間データで指定されるタイミングで M I D I データを処理するために、時間の進行に応じて値を更新する変数（現在時刻レジスタ）を用意し、その変数の値を時間データの値と対比することで、時間データで指定されたタイミングで M I D I データを処理するようにしている。

【 0 1 9 5 】

まず、ステップ 2 2 0 1 では、現在時刻レジスタの値を前回、更新してから 1 m s が経過したか否か判定する。前回の更新時のタイマ 1 0 2 の値とそのタイマ 1 0 2 の現在値の間に 1 m s に対応する値以上の差があった場合、判定は Y E S となり、ステップ 2 2 0 2 で現在時刻レジスタの値のインクリメントを行った後、ステップ 2 2 0 3 に移行する。そうでない場合には、判定は N O となってステップ 2 2 0 3 に移行する。

10

【 0 1 9 6 】

ステップ 2 2 0 3 では、受信音楽データバッファに格納した M I D I データのなかで次に処理すべき M I D I データを処理するタイミングとなったか否か判定する。その M I D I データに付加された時間データの値が現在時刻レジスタの値以下となった場合、判定は Y E S となり、ステップ 2 2 0 4 でその M I D I データを M I D I O U T 端子 1 1 3 に出力した後、ステップ 2 2 0 5 に移行する。そうでない場合には、判定は N O となってステップ 2 2 0 5 に移行する。

20

【 0 1 9 7 】

ステップ 2 2 0 5 では、受信器 1 0 7 を介してサーバに対する問い合わせを行い、自分宛に新たに送信されたデータがあればそれダウンロードさせる。続くステップ 2 2 0 6 では、受信器 1 0 7 から受け取った受信データ中の M I D I データ、及び時間データを受信音楽データバッファに格納する。そのようにして、受信器 1 0 7 の M I D I データの受信に対応する処理を行った後、ステップ 2 2 0 7 に移行する。

【 0 1 9 8 】

ステップ 2 2 0 7 では、ステップ 2 2 0 5 で受信した M I D I データがデータベース装置 2 0 0 2 から音楽情報として送信されてきた最初のデータか否か判定する。データベース装置 2 0 0 2 が演奏データの最初に位置している M I D I データを送信した場合、判定は Y E S となり、ステップ 2 2 0 8 でそれに付加されている時間データを現在時刻レジスタに代入した後、ステップ 2 2 0 9 に移行する。そうでない場合には、判定は N O となり、そのままステップ 2 2 0 9 に移行する。

30

【 0 1 9 9 】

このように、演奏装置 2 0 0 1 は、データベース装置 2 0 0 2 から最初に送信されてきた時間データを現在時刻レジスタに代入している。それにより、受信した M I D I データに付加された時間データの値を現在時刻レジスタの値と単に対比するだけで、その時間データで指定されたタイミングで M I D I データを処理できるようにしている。

【 0 2 0 0 】

ステップ 2 2 0 9 では、M I D I I N 端子 1 1 4 が鍵盤装置 1 3 0 から M I D I データを受信したか否か判定する。ユーザが鍵盤装置 1 3 0 を操作した場合、判定は Y E S となってステップ 2 2 1 0 に移行する。そうでない場合には、判定は N O となって一連の処理を終了する。

40

【 0 2 0 1 】

ステップ 2 2 1 0 では、M I D I I N 端子 1 1 4 が受信した M I D I データを受け取り、それを M I D I O U T 端子 1 1 3 に出力して音源装置 1 2 0 に処理させるとともに、その M I D I データに現在時刻レジスタの値を時間データとして付加して送信器 1 0 8 に送り、データベース装置 2 0 0 2 を送信先にして送信させる。それが終わった後、一連の処理を終了する。

50

## 【0202】

このようにして、演奏装置2001では、データベース装置2002から送信されてきた演奏データに対する処理、その装置2002への演奏データの送信を行う。それにより、上述した動作が実現されることになる。

## 【0203】

図23は、データベース装置2002に搭載されたCPU101が実行する音楽情報送受信処理の動作フローチャートである。次に、図23、更には図24を参照して、そのCPU101の制御動作について詳細に説明する。

## 【0204】

なお、このデータベース装置2002で実行される音楽情報送受信処理は、ユーザがスイッチ群115を介して、楽音発生装置がデータベース装置として機能することや送信の対象とする演奏データ、更には演奏装置2001及び受信装置2003～2005等の送信先を指定した後、その演奏データの送信の開始を指示した場合に、CPU101がROM103に格納されたプログラムを実行することで実現される。

## 【0205】

まず、ステップ2301では、イニシャライズを行う。それにより、現在時刻レジスタや演奏時刻レジスタ等の演奏データの送信に用いる変数にはその値のクリアを含む初期値の代入を行うとともに、RAM104に送信すべき演奏データの格納に用いる領域（以降、音楽データバッファと呼ぶ）の確保を行う。そのようなことを行った後、ステップ2302に移行する。

## 【0206】

ステップ2302では、ユーザが指定した演奏データをROM103から読み出し、それを音楽データバッファに格納する。続くステップ2303では、指定された送信先と演奏データの送受信、或いは送信を行うためのラベル2処理を実行する。それが終了した後、ステップ2304に移行する。

## 【0207】

ステップ2304では、音楽情報の送受信が終了したか否か判定する。演奏装置2001は、データベース装置2002から送信された演奏データによる演奏と合奏したユーザの演奏内容を演奏データとして送信することから、ユーザが指定した演奏データの送信の終了は演奏装置2001よりも各受信装置2003～2005のほうが後になる。このことから、その演奏データの各受信装置2003～2005への送信が終了してから予め定めた時間以上、演奏データを演奏装置2001が送信してこなかった、或いはユーザがスイッチ群115を介して演奏データの送信の中止を指示したような場合、判定はYESとなり、ここで一連の処理を終了する。そうでない場合には、判定はNOとなってステップ2303に戻り、ラベル2処理を再度実行する。

## 【0208】

上記ステップ2303とステップ2304とは処理ループを形成しており、そのステップ2304の判定がYESとなるまで繰り返し実行される。それにより、演奏装置2001との間の演奏データの送受信、各受信装置2003～2005への演奏データの送信が行われる。

## 【0209】

図24は、上記ステップ2303として実行されるラベル2処理の動作フローチャートである。次に、図24を参照して、ラベル2処理について詳細に説明する。

## 【0210】

データベース装置2002は、演奏装置2001と各受信装置2003～2005に同一の演奏データをMIDIデータ単位で順次送信するが、その送信のタイミングは少なからず送信に時間がかかることから異なる。同じMIDIデータを送信するタイミングは演奏装置2001のほうが早い。上記現在時刻レジスタ、及び演奏時刻レジスタは、このために用意した変数である。それらの変数を用いて演奏装置2001と各受信装置2003～2005にMIDIデータを送信するタイミングを別々に管理している。

10

20

30

40

50

## 【 0 2 1 1 】

先ず、ステップ 2 4 0 1 では、現在時刻レジスタの値を前回、更新してから 1 m s が経過したか否か判定する。前回の更新時のタイマ 1 0 2 の値とそのタイマ 1 0 2 の現在値の間に 1 m s に対応する値以上の差があった場合、判定は Y E S となり、現在時刻レジスタ、及び演奏時刻レジスタの各値のインクリメント（ステップ 2 4 0 2 及び 2 4 0 3 ）を行った後、ステップ 2 4 0 4 に移行する。そうでない場合には、判定は N O となり、次にステップ 2 4 0 4 に移行する。

## 【 0 2 1 2 】

ステップ 2 4 0 4 では、音楽データバッファに格納した M I D I データのなかで次に演奏装置 2 0 0 1 に送信すべき M I D I データを送信するタイミングとなったか否か判定する。その M I D I データに付加された時間データの値が現在時刻レジスタの値以下となった場合、判定は Y E S となり、ステップ 2 4 0 5 でその M I D I データと時間データを送信器 1 0 8 から演奏装置 2 0 0 1 に送信させた後、ステップ 2 4 0 6 に移行する。そうでない場合には、判定は N O となり、次にステップ 2 4 0 6 に移行する。

10

## 【 0 2 1 3 】

ステップ 2 4 0 6 では、受信器 1 0 7 を介してサーバに問い合わせを行い、自分宛にデータが送信されてきていればそれをダウンロードして、それが演奏装置 2 0 0 1 から送信された M I D I データか否か判定する。受信したデータに付加されている送信元が演奏装置 2 0 0 1 を示していた場合、判定は Y E S となってステップ 2 4 0 7 に移行する。そうでない場合には、判定は N O となってステップ 2 4 1 0 に移行する。

20

## 【 0 2 1 4 】

ステップ 2 4 0 7 では、受信器 1 0 7 が受信した M I D I データが演奏装置 2 0 0 1 から音楽情報として送信されてきた最初のデータか否か判定する。演奏装置 2 0 0 1 への演奏データの送信を開始した後、ユーザがその鍵盤装置 1 3 0 への操作を開始した場合、最初の M I D I データが受信されることから、判定は Y E S となり、ステップ 2 4 0 8 でその M I D I データに付加されている時間データを演奏時刻レジスタに代入した後、ステップ 2 4 0 9 に移行する。そうでない場合には、判定は N O となってステップ 2 4 0 9 に移行する。そのステップ 2 4 0 9 では、送信器 1 0 8 に、演奏装置 2 0 0 1 から受信した M I D I データ、及び時間データを各受信装置 2 0 0 3 ~ 2 0 0 5 に送信させる。

## 【 0 2 1 5 】

このように、データベース装置 2 0 0 2 は、演奏装置 2 0 0 1 から最初に送信されてきた時間データを演奏時刻レジスタに代入している。その時間データは、データベース装置 2 0 0 2 から最初に送信された時間データの値を初期値にして計時した時刻を表すデータである。このため、後述するように、演奏時刻レジスタの値を時間データと単に対比して送信するタイミングを特定するだけで、ユーザが指定した演奏データ、及び演奏装置 2 0 0 1 から受信した演奏データを各受信装置 2 0 0 3 ~ 2 0 0 5 に M I D I データ単位で送信するタイミングのズレが回避される。それにより、それらの演奏データの再生は同期して各受信装置 2 0 0 3 ~ 2 0 0 5 で行われることになる。

30

## 【 0 2 1 6 】

ステップ 2 4 0 9 に続くステップ 2 4 1 0 では、音楽データバッファに格納した M I D I データのなかで次に各受信装置 2 0 0 3 ~ 2 0 0 5 に送信すべき M I D I データを送信するタイミングとなったか否か判定する。その M I D I データに付加された時間データの値が演奏時刻レジスタの値以下となった場合、判定は Y E S となり、ステップ 2 4 1 1 でその M I D I データと時間データを送信器 1 0 8 から各受信装置 2 0 0 3 ~ 2 0 0 5 に送信させた後、一連の処理を終了する。そうでない場合には、判定は N O となり、ここで一連の処理を終了する。

40

## 【 0 2 1 7 】

図 2 5 は、各受信装置 2 0 0 3 ~ 2 0 0 5 にそれぞれ搭載された C P U 1 0 1 が実行する音楽情報受信処理の動作フローチャートである。次に、図 2 5 を参照して、各受信装置 2 0 0 3 ~ 2 0 0 5 にそれぞれ搭載された C P U 1 0 1 の動作について詳細に説明する。

50

## 【0218】

なお、その音楽情報受信処理は、音楽情報送受信装置が受信装置として機能することをスイッチ群115によりユーザが指定した場合に、CPU101がROM103に格納されたプログラムを実行することで実現される。

## 【0219】

先ず、ステップ2501では、イニシャライズを行う。それにより、各種変数にはその値のクリアを含めて予め定められた値を代入し、受信した演奏データを一時的に保持しておく領域（以降、受信音楽データバッファと呼ぶ）をRAM104に確保する。続くステップ2502では、データベース装置2002から送信された演奏データを受信してそれを再生するためのラベル3処理を実行する。ステップ2503にはその後に移行する。

10

## 【0220】

ステップ2503では、音楽情報の受信が終了したか否か判定する。データベース装置2002から予め定めた時間以上、演奏データが送信されなかった、或いはユーザがスイッチ群115を介して演奏データの受信の中止を指示したような場合、判定はYESとなり、ここで一連の処理を終了する。そうでない場合には、判定はNOとなってステップ2502に戻り、ラベル3処理を実行する。

## 【0221】

上記ステップ2502とステップ2503は処理ループを形成しており、そのステップ2503の判定がYESとなるまで繰り返し実行される。それにより、データベース装置2002から送信されたMIDIデータを順次受信して処理することによる自動演奏を行うようになっている。

20

## 【0222】

図26は、上記ステップ2502として実行されるラベル3処理の動作フローチャートである。次に、図26を参照して、ラベル3処理について詳細に説明する。

## 【0223】

先ず、ステップ2601では、現在時刻レジスタの値を前回、更新してから1msが経過したか否か判定する。前回の更新時のタイマ102の値とそのタイマ102の現在値の間に1msに対応する値以上の差があった場合、判定はYESとなり、ステップ2602で現在時刻レジスタの値のインクリメントを行った後、ステップ2603に移行する。そうでない場合には、判定はNOとなってステップ2603に移行する。

30

## 【0224】

ステップ2603では、受信音楽データバッファに格納したMIDIデータのなかで次に処理すべきMIDIデータを処理するタイミングとなったか否か判定する。そのMIDIデータに付加された時間データの値が現在時刻レジスタの値以下となった場合、判定はYESとなり、ステップ2604でそのMIDIデータをMIDI OUT端子113に出力した後、ステップ2605に移行する。そうでない場合には、判定はNOとなってステップ2605に移行する。

## 【0225】

ステップ2605では、受信器107を介してサーバに問い合わせを行い、自分宛にデータが送信されてきたか否か判定する。データベース装置2002がMIDIデータを自分宛に送信した場合、判定はYESとなり、ステップ2606でそのデータをダウンロードさせ、それを受信器107から受け取って受信音楽データバッファに格納した後、ステップ2607に移行する。そうでない場合には、判定はNOとなり、ここで一連の処理を終了する。

40

## 【0226】

ステップ2607では、直前に実行したステップ2606で受信音楽データバッファに格納したデータがデータベース装置2002から音楽情報として送信されてきた最初のデータか否か判定する。そのデータの前にデータベース装置2002からMIDIデータを受信していなかった場合、判定はYESとなり、ステップ2608でそれに付加されている時間データを現在時刻レジスタに代入した後、一連の処理を終了する。そうでない場合に

50

は、判定はNOとなり、ここで一連の処理を終了する。

【0227】

このように、各受信装置2003～2005は、データベース装置2002から最初に送信されてきた時間データを現在時刻レジスタに代入している。それにより、受信したMIDIデータに付加された時間データの値を現在時刻レジスタの値と単に対比するだけで、ユーザが指定した演奏データが演奏装置2001で生成された演奏データかに係わらず、それを構成するMIDIデータを適切なタイミングで処理できるようにしている。その結果、各受信装置2003～2005では、演奏装置2001で行われた合奏が忠実に再現される。

【0228】

なお、第3の実施の形態では、データベース装置2002が演奏装置2001と受信装置2003～2005の2つに分けて異なる音楽情報を送信するようにしているが、それ以上に分けて異なる音楽情報を送信できるようにしても良い。その音楽情報の内容については、パートや音色等に着目して指定できるようにしても良い。

【0229】

そのデータベース装置2002は、演奏装置2001から音楽情報を受信するようになっているが、外部装置から音楽情報を受信せず、単に音楽情報を外部装置に送信するものとして使用できるようにしても良い。そのような使用方法に限定する場合には、受信器107は備えなくても良い。

<第4の実施の形態>

合奏、或いは競演等(これらをまとめてセッションと表現する)は、複数の演奏者が一個所に集まらなければ行うことができず、場所的な制約がある。第4の実施の形態は、そのような制約を解消して、複数の演奏者が一個所に集まらなくてもセッションを行えるようにしたものである。

【0230】

第4の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の構成は、上記第1の実施の形態におけるそれ(図1参照)と基本的に同じである。このことから、第4の実施の形態でも、上記第2及び第3の実施の形態と同様に、第1の実施の形態の説明で用いた符号を基本的にはそのまま流用して、図1を参照しつつ説明することにする。

【0231】

図27は、第4の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の利用方法を説明する図である。先ず、図27を参照して、その利用方法、及びそれを可能にさせる動作について説明する。

【0232】

図27中の2701～2703は、例えば、それぞれが図1に示す構成を有する楽音発生装置である。矢印は、音楽情報がどこからどこに送信されるのかを示している。即ち、楽音発生装置2701は楽音発生装置2702及び2703と音楽情報の送受信を行い、楽音発生装置2702は楽音発生装置2701の他に楽音発生装置2703と音楽情報の送受信を行うことを示している。第1～第3の実施の形態と同様に、それらの装置2701～2703はLANに接続されている。

【0233】

比較的遠く離れた位置にいる演奏者に音楽情報を送る場合、少なからず遅延が生じることから、実時間処理では原理的に演奏者間の演奏のズレを回避することができない。自身の演奏と、聴こえてくる他の演奏者の演奏との間のタイミング的なズレが生じている演奏者が必ず存在してしまう。

【0234】

しかし、実際のセッションには、例えば楽譜に忠実な演奏から始めて、各演奏者が演奏(テイク)を繰り返す毎にそれぞれの演奏内容を調整していくといったように、同じ曲の演奏を繰り返し行うこともある。第4の実施の形態では、実時間処理では演奏者間の演奏のズレを回避できないことから、テイクを繰り返すセッションを行う場合に、以下のように

10

20

30

40

50

して各テイクをより適切に行えるようにしている。ここでは、便宜的に、図 27 に示す各装置 2701 ~ 2703 が持つ演奏データに対応する曲でセッションを行う場合を例にとって説明する。また、各装置 2701 ~ 2703 の動作は基本的には同じであることから、装置 2701 に注目して説明することにする。

【0235】

上記演奏データは複数のパートを持つ各演奏者の演奏意図を含まないデータである。その演奏データは、スイッチ群 115 への操作によって楽音発生装置 2701 のユーザ（演奏者）に指定されると、CPU 101 によって ROM 103 から読み出されて、RAM 104 に確保した領域（以降、音楽トラックバッファと呼ぶ）に格納される。

【0236】

楽音発生装置 2701 のユーザ（演奏者）は、例えば、自身が演奏するパートの発音のミュートを指定して演奏データの再生（マイナスワン演奏）を行わせ、その演奏に合わせて所望のパートの演奏を行う。CPU 101 は、そのパートの演奏内容を示す MIDI データを MIDI IN 端子 114 から受け取り、その MIDI データにその処理タイミングを指定する時間データを付加し、それらを音楽情報として随時、各装置 2702 及び 2703 に送信する。

【0237】

演奏データ中の MIDI データには、演奏の開始時を基準として処理タイミングを表す時間データを付加するようになっている。このことから、送信する MIDI データには、演奏データの再生開始を基準として計時した時刻を時間データとして付加している。その時刻の計時には、変数である演奏時刻レジスタを用いている。

【0238】

各楽音発生装置 2702 及び 2703 でも同様に、そのユーザが行った演奏内容を示す音楽情報を他の装置に送信する。楽音発生装置 2701 は、楽音発生装置 2702 或いは 2703 から送信された音楽情報を受信すると、それを RAM 104 に確保した領域（以降、受信音楽データバッファと呼ぶ）に装置（パート（トラック）に対応する）別に格納する。それにより、そのバッファに、各楽音発生装置 2702 及び 2703 から送信された各パートの演奏データを格納する。

【0239】

受信音楽データバッファに格納した演奏データは、ユーザの演奏の終了を待って、音楽トラックバッファに格納した演奏データ中のそのパートに対応しているデータと入れ換える。音楽トラックバッファに格納した演奏データに、各演奏者（ここでは二人）が直前に行った演奏の内容を取り込む更新を行う。それにより、楽音発生装置 2701 のユーザが、楽音発生装置 2702 及び 2703 の各ユーザが直前に行った演奏との合奏を行えるようにしている。それらの装置 2702 及び 2703 は MIDI データに時間データを付加して送信する。このため、更新後の演奏データを再生させての合奏を行っても、それらの装置 2702 及び 2703 で直前に行われた演奏との間のタイミング的なズレは回避される。その結果、楽音発生装置 2701 のユーザは、他の演奏者（他の装置 2702 及び 2703 の各ユーザ）がその場所に居るような感覚で快適にテイクを繰り返すことができるようになる。

【0240】

なお、上述したような演奏データの送受信、及び演奏データの更新は、ユーザがスイッチ群 115 を操作して所定のモード（以降、便宜的にセッションモードと呼ぶ）を設定した場合に行われる。また、本実施の形態では、上記演奏データの更新を容易に行えるように、パート単位に分けて編成された演奏データを採用し、それを ROM 103 に格納させている。

【0241】

次に、上記各楽音発生装置 2701 ~ 2703 の動作を実現させるそれに搭載された CPU 101 の制御動作について、図 28 ~ 図 33 を参照して詳細に説明する。それらの装置 2701 ~ 2703 の動作は基本的に同じであることから、ここでも装置 2701 に着目

10

20

30

40

50

して説明することにする。

【0242】

図28は、楽音発生装置2701に搭載されたCPU101が実行する音楽情報送受信処理の動作フローチャートである。始めに、図28を参照して、その処理の内容について詳細に説明する。

【0243】

なお、その音楽情報送受信処理は、楽音発生装置2701のユーザが、スイッチ群115を操作して、上記セッションモードを設定するとともに、再生させる演奏データ(曲)、及び音楽情報(ここでは演奏データ)の送信先を各々指定した場合に、CPU101がROM103に格納されたプログラムを実行することで実現される。指定する演奏データ(曲)は、セッションを行う演奏者(ユーザ)間で予め決めておくことで、各装置2701~2703間で統一させることができる。

10

【0244】

先ず、ステップ2801では、イニシャライズを行う。それにより、演奏時刻レジスタの値のクリアを含めて各種変数に予め定められた値を代入し、ROM103から読み出した演奏データの格納に用いる音楽トラックバッファ、及び受信した演奏データを一時的に保持しておく受信音楽データバッファ等をRAM104に確保する。続くステップ2802では、ユーザが指定した演奏データをROM103から読み出して音楽トラックバッファに格納する。その後、ステップ2803に移行する。

【0245】

ステップ2803では、上記演奏時刻レジスタの値を随時、更新しつつ、他の装置2702及び2703との間で演奏データの送受信や演奏データの更新、更には演奏データの再生を行うためのラベル1処理を実行する。それを実行すると、ステップ2804に移行する。

20

【0246】

ステップ2804では、ユーザがセッションの終了を指示したか否か判定する。ユーザがスイッチ群115を操作してセッションモードの解除、或いは演奏データの再生の終了を指示したような場合、判定はYESとなって一連の処理を終了する。そうでない場合には、判定はNOとなってステップ2803に戻ってラベル1処理を実行する。

【0247】

上記のように、ステップ2803とステップ2804は処理ループを形成しており、ステップ2804の判定がYESとなるまでの間、繰り返し実行される。それにより、他の装置2702や2703との間の演奏データの送受信、演奏データを更新しての再生が行われることになる。

30

【0248】

図29は、上記ステップ2803として実行されるラベル1処理の動作フローチャートである。次に、図29を参照して、ラベル2処理について詳細に説明する。

【0249】

先ず、ステップ2901では、前回、演奏時刻レジスタの値を更新(インクリメント)してから1ms経過したか否か判定する。その更新を行ったときのタイマ102の値と現在のその値との間に1msに対応する値以上の差があった場合、判定はYESとなり、ステップ2902でその演奏時刻レジスタの値をインクリメントした後、ステップ2903に移行する。そうでない場合には、判定はNOとなり、次にステップ2903に移行する。

40

【0250】

ステップ2903では、他の装置2702及び2703との間で演奏データの送受信や演奏データの更新、更には演奏データの再生を行うためのラベル2処理を実行する。それを実行した後、一連の処理を終了する。

【0251】

図30は、上記ステップ2903として実行されるラベル2処理の動作フローチャートで

50

ある。次に、図30を参照して、ラベル2処理について詳細に説明する。

【0252】

各楽音発生装置2701～2703間でのデータの送受信は、サーバを介して行われる。このことから、まず、ステップ3001でサーバに問い合わせを行うことにより、自分宛に送信されたデータがあるか否か判定する。そのサーバが楽音発生装置2702、或いは2703から楽音発生装置2701に宛てたデータを新たに保持していた場合、判定はYESとなってステップ3002に移行する。そうでない場合には、判定はNOとなってステップ3005に移行する。

【0253】

ステップ3002では、受信器107を介してサーバからデータをダウンロードする。続くステップ3003では、そのダウンロードしたデータを、それぞれ、その送信元により特定される受信音楽データバッファ内の領域に格納する。その後、ステップ3004に移行してサーバに対する問い合わせを行い、サーバから更にダウンロードすべきデータがないか否か判定する。その問い合わせの結果、サーバから更にダウンロードすべきデータのないことが確認された場合、判定はYESとなってステップ3005に移行する。そうでない場合には、判定はNOとなってステップ3002に戻り、ダウンロードすべきデータのダウンロードを行う。

10

【0254】

このようにして、他の楽音発生装置2702或いは2703から送信されたデータを受信した後のステップ3005では、音楽トラックバッファに格納された演奏データを再生したり、ユーザが行った演奏の内容を示すMIDIデータを各楽音発生装置2702及び2703に随時送信するといったことを行うためのラベル3処理を実行する。それが終了した後の一連の処理を終了する。

20

【0255】

図31は、上記ステップ3005として実行されるラベル3処理の動作フローチャートである。次に、図31を参照して、そのラベル3処理について詳細に説明する。

【0256】

上述したように、本実施の形態では、音楽トラックバッファに格納した演奏データの更新を容易に行えるように、パートに対応するトラック別に分けて構成されている演奏データを採用している。このことから、演奏データの再生は、トラック単位で再生の対象とするトラックを変更しながら行っている。再生の対象とするトラックは、変数であるトラックアドレスレジスタで管理している。このようにして行う演奏データの再生は、ステップ3101～3106の処理を実行することで実現される。

30

【0257】

まず、ステップ3101では、トラックアドレスレジスタの値で指定されるトラックの演奏データ中で次の処理対象のMIDIデータに付加された時間データの値が、演奏時刻レジスタの値以上か否か判定する。その時間データで指定される処理タイミングとなった場合、判定はYESとなってステップ3102に移行し、それが付加されたMIDIデータをMIDI OUT端子113に出力するとともに、処理対象をそのMIDIデータの次に位置しているMIDIデータに変更した後、ステップ3103に移行する。そうでない場合には、判定はNOとなり、次にステップ3103を実行する。

40

【0258】

そのステップ3103では、現在再生の対象としているトラックで処理の対象とすべきMIDIデータがもうないか否か判定する。上記ステップ3101の判定がNO、或いはそのトラックの演奏データ中の全てのMIDIデータを処理していた場合、判定はYESとなり、次にステップ3104でトラックアドレスレジスタの値をインクリメントした後、ステップ3105に移行する。そうでない場合には、判定はNOとなってステップ3101に戻り、次の処理対象のMIDIデータの処理タイミングについての判定を行う。

【0259】

ステップ3105では、再生の対象とするトラックがもうないか否か判定する。トラック

50

アドレスレジスタの値がトラック番号の最大値よりも大きい場合、言い換えれば全てのトラックの演奏データを対象に現時点での処理が終了した場合、判定はYESとなり、ステップ3106でトラックアドレスレジスタの値をクリア(0を代入)した後、ステップ3107に移行する。そうでない場合には、判定はNOとなり、ステップ3101に戻って新たに対象になったトラックの演奏データを再生するための処理を行う。

【0260】

ステップ3107では、MIDI IN端子114が鍵盤装置130からMIDIデータを入力したか否かが判定する。MIDI IN端子114がMIDIデータを受信した旨を示す信号を出力していた場合、判定はYESとなり、ステップ3108でMIDI IN端子114からMIDIデータを受け取って一時バッファに格納し、更にそれをMIDI OUT端子113に出力した後、ステップ3109に移行する。そうでない場合には、判定はNOとなり、次にステップ3109を実行する。なお、上記一時バッファは、鍵盤装置130が出力したMIDIデータを一時、保持しておくために、RAM104に予め確保した領域である。

10

【0261】

ステップ3109では、ステップ3108で一時バッファに格納したMIDIデータの送信や演奏データの再生の終了等に対処するためのラベル4処理を実行する。それが終了した後、一連の処理を終了する。

【0262】

図32は、そのステップ3109として実行されるラベル4処理の動作フローチャートである。次に、ラベル4処理について、図32を参照して詳細に説明する。

20

【0263】

CPU101は、ユーザがスイッチ群115を介して指定(入力)した送信先(アドレス)のリスト(以降、送信先リストと呼ぶ)をRAM104に格納する。鍵盤装置130から入力したMIDIデータの送信は、そのリストを参照して行われる。送信先リストのなかでデータの送信対象とする送信先は、変数である送信アドレスレジスタにより管理している。

【0264】

まず、ステップ3201では、一時バッファに格納されたMIDIデータを読み出してそれに時間データとして演奏時刻レジスタの値を付加し、その時間データを付加したMIDIデータを送信アドレスレジスタの値で指定される送信先リスト中の送信先に送信する。なお、その送信は、送信先のアドレスを格納したヘッダを送信対象のデータ(MIDIデータ、及び時間データ)に付加し、それを送信器108に出力することで行われる。

30

【0265】

上記演奏時刻レジスタは、演奏データ中のMIDIデータの処理タイミングを特定するために、時刻の計時に用いている変数である。そのレジスタの値を時間データとしてMIDIデータに付加することにより、楽音発生装置2701で行ったパートの演奏が、送信先で他のパートの演奏と同期して再現されることになる。それにより、各演奏者がそれぞれ異なる場所に居ても、一個所に集まっているような感覚でテイク(セッション)を繰り返すことができる。

40

【0266】

ステップ3201に続くステップ3202では、送信アドレスレジスタの値をインクリメントする。それが終了すると、ステップ3203に移行して、データを送信していない送信先がないか否かが判定する。送信アドレスレジスタの値が送信先リストに載っている送信先の番号の最大値よりも大きい場合、それは送信先リストに載っている全ての送信先にデータを送信したことを意味することから、判定はYESとなり、ステップ3204で一時バッファ内のMIDIデータを全て消去するとともに、送信アドレスレジスタの値をクリア(0を代入)した後、ステップ3205に移行する。そうでない場合には、判定はNOとなってステップ3201に戻り、次の送信先にデータを送信する。

【0267】

50

ラベル4処理の実行時に一時バッファにMIDIデータが格納されていない場合もある。その場合には、上述のステップ3201～3204は実際上、実行することなくステップ3205に移行することになる。

【0268】

ステップ3205では、音楽トラックバッファに格納した演奏データ中に処理していないMIDIデータが存在するか否か判定する。その演奏データの再生が完了していない場合、判定はYESとなり、ここで一連の処理を終了する。そうでない場合には、判定はNOとなってステップ3206に移行する。

【0269】

ステップ3206では、他の楽音発生装置3207及び3203から随時、受信した演奏データを音楽トラックバッファに格納した演奏データ中にコピーするラベル5処理を実行する。それが終了した後、一連の処理を終了する。

10

【0270】

このように、本実施の形態では、ユーザが演奏データの再生による演奏との合奏を行うとの前提から、その演奏データの再生の完了を待って、その更新を行うようにしている。その更新を行うことにより、演奏者各自が自身の演奏を他の演奏者の演奏と合わせていくことが可能となる。

【0271】

図33は、上記ステップ3206として実行されるラベル5処理の動作フローチャートである。次に、ラベル5処理について、図33を参照して詳細に説明する。

20

【0272】

上述したように、他の楽音発生装置2702及び2703から受信した演奏データは、受信音楽データバッファに装置(パート)別に格納している。このことから、音楽トラックバッファに格納されている演奏データ中への受信した演奏データのコピーは、装置(パート)単位で順次、行うようになっている。そのコピーの対象としている演奏データの管理は、その演奏データは送信先リストに載っている各送信先から送信されてくるのを前提としていることから、上記送信アドレスレジスタを用いて行っている。

【0273】

まず、ステップ3301では、音楽トラックバッファの演奏データ中で送信アドレスレジスタの値により特定されるトラックに、そのトラックに対応する受信音楽データバッファ内の演奏データを書き込む(上書きする)。それにより、1パート分の演奏データをコピーする。なお、当然のことながら、そのコピーは、受信音楽データバッファ中にコピーの対象となる演奏データが無い場合には行わない。

30

【0274】

ステップ3301に続くステップ3302では、送信アドレスレジスタの値をインクリメントする。それを行うと、次にステップ3303に移行して、受信音楽データバッファ中に格納されている演奏データを全てコピーしたか否か判定する。その送信アドレスレジスタの値が送信先リストに載っている送信先の番号の最大値よりも大きい場合、それは送信先リストに載っている送信先から受信した演奏データを全てコピーしたことを意味することから、判定はYESとなってステップ3304に移行する。そうでない場合には、判定はNOとなってステップ3301に戻り、受信音楽データバッファ中のコピー対象を替えて演奏データを音楽トラックバッファの演奏データ中にコピーする。

40

【0275】

上記のように、ステップ3301～3303は処理ループを形成している。その処理ループを実行することにより、音楽トラックバッファに格納された演奏データの更新がパート(装置)単位で行われる。

【0276】

ステップ3303の判定がYESとなった後は、送信アドレスレジスタ、及び演奏時刻レジスタの値をそれぞれクリア(0を代入)する(ステップ3304、3305)。そのクリアを行った後、一連の処理を終了する。

50

## 【 0 2 7 7 】

ラベル 5 処理が終了すると、図 2 8 に示す音楽情報送受信処理でステップ 2 8 0 3 として実行されるラベル 1 処理も終了することになる。そのステップ 2 8 0 3 に続くステップ 2 8 0 4 の判定が N O の場合、即ちユーザが演奏の中止を指示しないような場合、演奏を行った後、ユーザに直ちに次の演奏を開始させることになる。しかし、そのようにすることは、各演奏者の都合を無視することになるから望ましくない。このため、演奏データの再生が終了した場合には、ステップ 2 8 0 4 からステップ 2 8 0 3 への移行は、演奏データの再生が終了してから所定の時間が経過するのを待った後、或いは全ての演奏者が用意できたことを示す操作を行うのを待った後、といったように、予め定めた条件が満たされた後に行うことが望ましい。そのようにした場合には、各演奏者に一個所に集まっていると

10

## 【 0 2 7 8 】

なお、第 4 の実施の形態では、3 人の演奏者がセッションを行う場合を例にとって説明を行っているが、当然のことながら、その人数は 3 人に限定されるものではない。また、セッションを繰り返す毎に演奏データの更新を行うようになっているが、その更新は、例えば予め定めた時間毎といったように、他の基準で行うようにしても良い。そのようにした場合には、各自に演奏の調整のための練習時間を与えることができる。

## 【 0 2 7 9 】

また、第 4 の実施の形態では、複数の装置と音楽情報の送受信を行えるようになっているが、1 つの装置とのみそれを行えるようにしても良い。更には、自分の力量に応じて、音楽情報の送信、及び受信の一方のみを選択できるようにしても良い。このように、様々な変形を行うことができる。

20

## 【 0 2 8 0 】

本実施の形態（第 1 ～ 第 4 の実施の形態）は、楽音発生装置に搭載される装置として実現させたものであるが、本発明を適用した装置の搭載対象は、電子楽器や各種コントローラ、シーケンサ等の楽音発生装置に限定されるものではない。パーソナルコンピュータ等にも搭載させることができる。その本発明を適用した装置には、当然のことながら、音楽情報の送受信を行う機能を必ずしも搭載させる必要はなく、想定する利用環境に応じて、音楽情報を送信する機能、及び受信する機能の何れか一方だけを搭載するようにしても良い。

30

## 【 0 2 8 1 】

パーソナルコンピュータへの搭載は、基本的には上記した動作を実現させるためのプログラムをロードすることで行うことができる。そのプログラムは、C D - R O M やフロッピーディスクといった携帯性に優れた記録媒体に記録して配布しても良く、何らかの通信回線を介して配信しても良い。

## 【 0 2 8 2 】

## 【 発明の効果 】

以上説明したように、本発明は、例えばパート別に音楽情報を分けて送信するといったように、複数の外部装置に送信する音楽情報の内容を外部装置単位で異ならせつつ、音楽情報をそれが有する時間データで指定されるタイミングに基づいて送信する。このため、ユーザは音楽情報を受信させる各装置（それを搭載した装置）をより所望する形で使用することができる。

40

## 【 図面の簡単な説明 】

【 図 1 】 第 1 の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の回路構成図である。

【 図 2 】 音楽情報送信処理の動作フローチャートである。

【 図 3 】 ラベル 1 処理の動作フローチャートである。

【 図 4 】 ラベル 2 処理の動作フローチャートである。

【 図 5 】 音楽情報受信処理の動作フローチャートである。

【 図 6 】 ラベル 3 処理の動作フローチャートである。

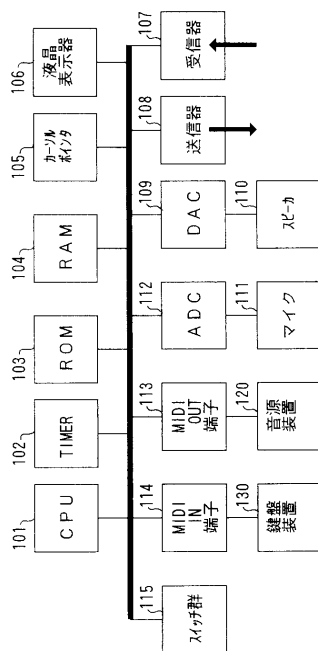
50

- 【図 7】ラベル 4 処理の動作フローチャートである。
- 【図 8】ラベル 8 処理の動作フローチャートである。
- 【図 9】ラベル 5 処理の動作フローチャートである。
- 【図 10】ラベル 9 処理の動作フローチャートである。
- 【図 11】ラベル 6 処理の動作フローチャートである。
- 【図 12】ラベル 7 処理の動作フローチャートである。
- 【図 13】第 2 の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の利用方法を説明する図である。
- 【図 14】音楽情報送信処理の動作フローチャートである（第 2 の実施の形態）。
- 【図 15】ラベル 1 処理の動作フローチャートである（第 2 の実施の形態）。 10
- 【図 16】音楽情報受信処理の動作フローチャートである（第 2 の実施の形態）。
- 【図 17】ラベル 2 処理の動作フローチャートである（第 2 の実施の形態）。
- 【図 18】ラベル 3 処理の動作フローチャートである（第 2 の実施の形態）。
- 【図 19】ラベル 4 処理の動作フローチャートである（第 2 の実施の形態）。
- 【図 20】第 3 の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の利用方法を説明する図である。
- 【図 21】演奏装置で実行される音楽情報送受信処理の動作フローチャートである（第 3 の実施の形態）。
- 【図 22】ラベル 1 処理の動作フローチャートである（第 3 の実施の形態）。
- 【図 23】データベース装置で実行される音楽情報送受信処理の動作フローチャートである（第 3 の実施の形態）。 20
- 【図 24】ラベル 2 処理の動作フローチャートである（第 3 の実施の形態）。
- 【図 25】受信装置で実行される音楽情報受信処理の動作フローチャートである（第 3 の実施の形態）。
- 【図 26】ラベル 3 処理の動作フローチャートである（第 3 の実施の形態）。
- 【図 27】第 4 の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の利用方法を説明する図である。
- 【図 28】音楽情報送受信処理の動作フローチャートである（第 4 の実施の形態）。
- 【図 29】ラベル 1 処理の動作フローチャートである（第 4 の実施の形態）。
- 【図 30】ラベル 2 処理の動作フローチャートである（第 4 の実施の形態）。 30
- 【図 31】ラベル 3 処理の動作フローチャートである（第 4 の実施の形態）。
- 【図 32】ラベル 4 処理の動作フローチャートである（第 4 の実施の形態）。
- 【図 33】ラベル 5 処理の動作フローチャートである（第 4 の実施の形態）。
- 【符号の説明】
- 1 0 0、1 3 0 1 ~ 1 3 0 4、2 7 0 1 ~ 2 7 0 3 楽音発生装置
- 1 0 1 C P U
- 1 0 2 タイマ
- 1 0 3 R O M
- 1 0 4 R A M
- 1 0 5 カーソルポインタ 40
- 1 0 6 液晶表示器
- 1 0 7 受信器
- 1 0 8 送信器
- 1 0 9 D A コンバータ
- 1 1 0 スピーカ
- 1 1 1 マイク
- 1 1 2 A D コンバータ
- 1 1 3 M I D I O U T 端子
- 1 1 4 M I D I I N 端子
- 1 1 5 スイッチ群 50

- 2001 演奏装置
- 2002 データベース装置
- 2003 ~ 2005 受信装置

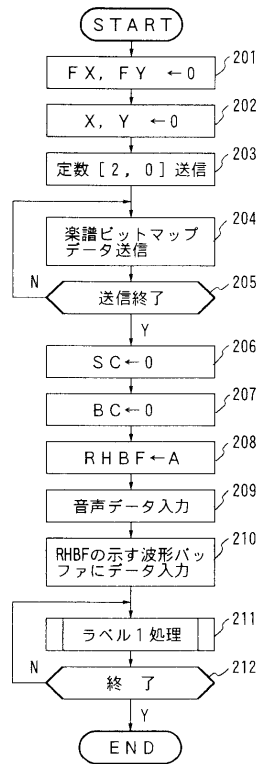
【 図 1 】

第1の実施の形態による音楽情報受信装置を搭載した楽音発生装置の回路構成図



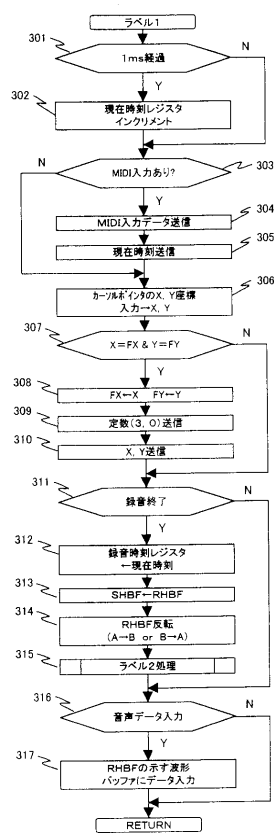
【 図 2 】

音楽情報送信処理の動作フローチャート



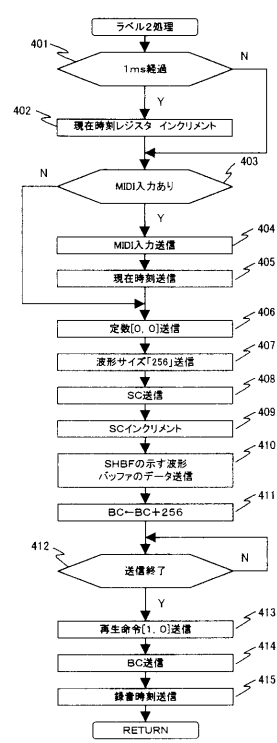
【 図 3 】

ラベル1処理の動作フローチャート



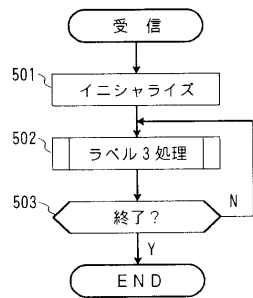
【 図 4 】

ラベル2処理の動作フローチャート



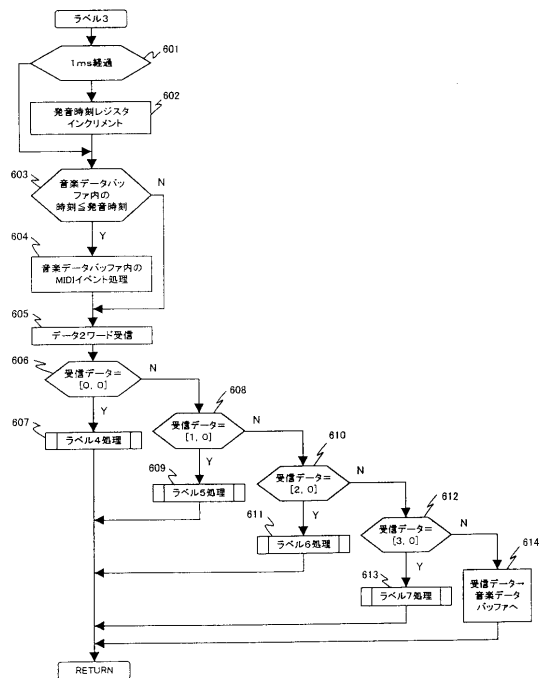
【 図 5 】

音楽情報受信処理の動作フローチャート



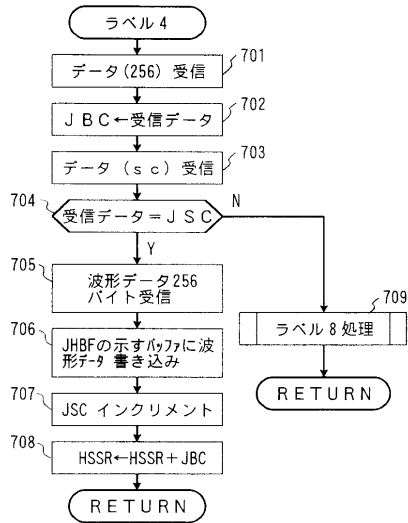
【 図 6 】

ラベル3処理の動作フローチャート



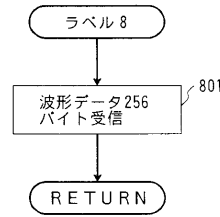
【 図 7 】

ラベル4 処理の動作フローチャート



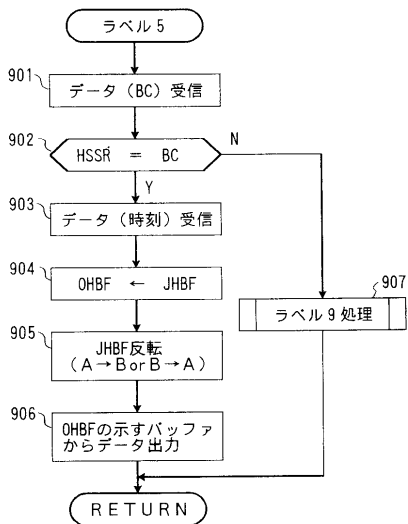
【 図 8 】

ラベル8 処理の動作フローチャート



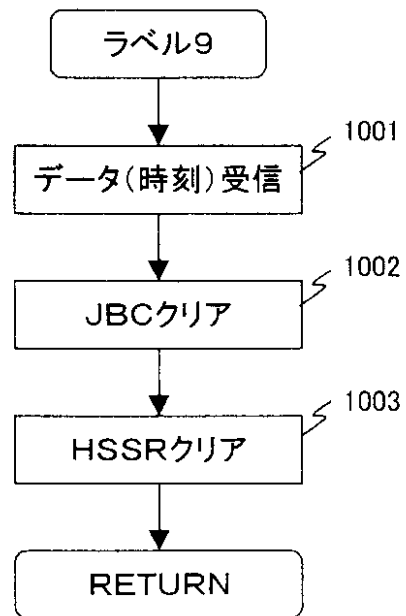
【 図 9 】

ラベル5 処理の動作フローチャート



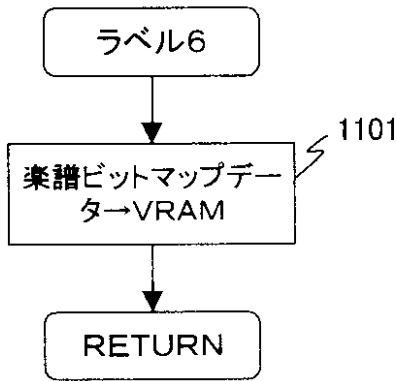
【 図 10 】

ラベル9 処理の動作フローチャート



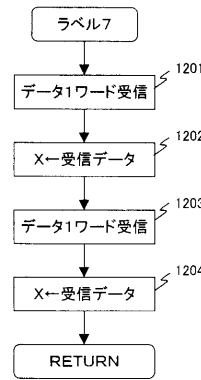
【 図 1 1 】

ラベル6処理のフローチャート



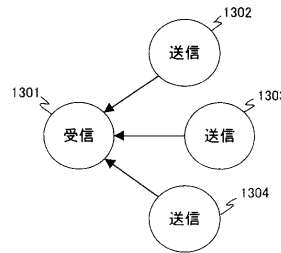
【 図 1 2 】

ラベル7処理の動作フローチャート



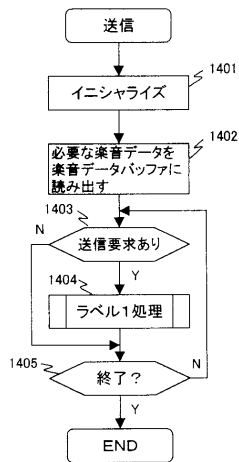
【 図 1 3 】

第2の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の利用方法を説明する図



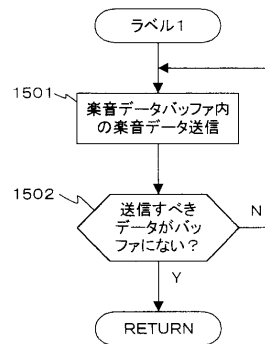
【 図 1 4 】

音楽情報送信処理の動作フローチャート (第2の実施の形態)



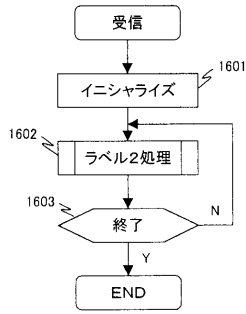
【 図 1 5 】

ラベル1処理の動作フローチャート (第2の実施の形態)



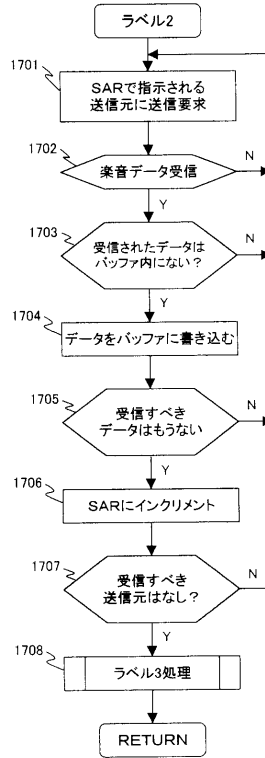
【 図 1 6 】

音楽情報受信処理の動作フローチャート  
(第2の実施の形態)



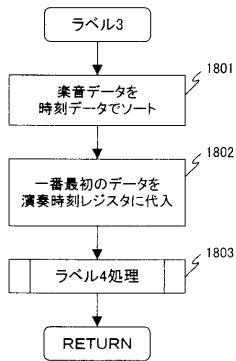
【 図 1 7 】

ラベル2処理の動作フローチャート  
(第2の実施の形態)



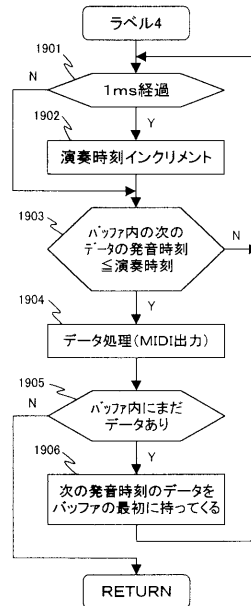
【 図 1 8 】

ラベル3処理の動作フローチャート  
(第2の実施の形態)



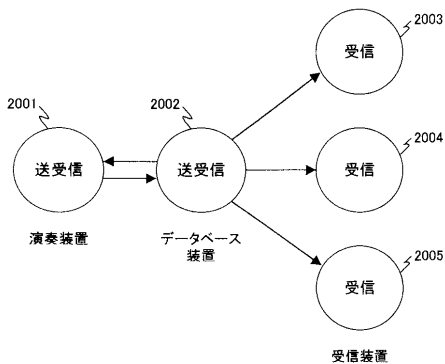
【 図 1 9 】

ラベル4処理の動作フローチャート  
(第2の実施の形態)



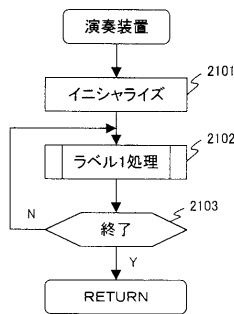
【図20】

第3の実施の形態による音楽情報送受信装置を搭載した楽音発生装置の利用方法を説明する図



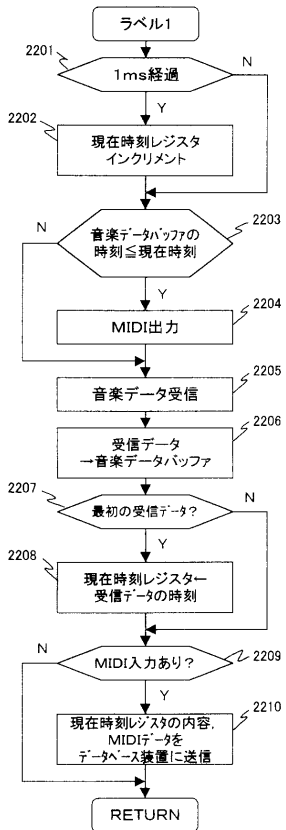
【図21】

演奏装置で実行される音楽情報送受信処理の動作フローチャート(第3の実施の形態)



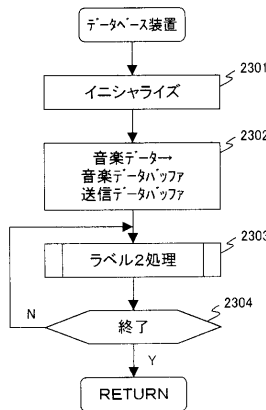
【図22】

ラベル1処理の動作フローチャート(第3の実施の形態)



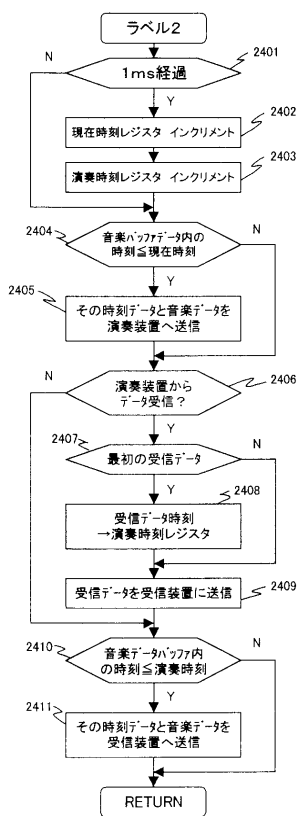
【図23】

データベース装置で実行される音楽情報送受信装置の動作フローチャート(第3の実施の形態)



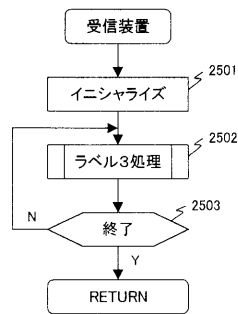
【 図 2 4 】

ラベル2処理の動作フローチャート  
(第3の実施の形態)



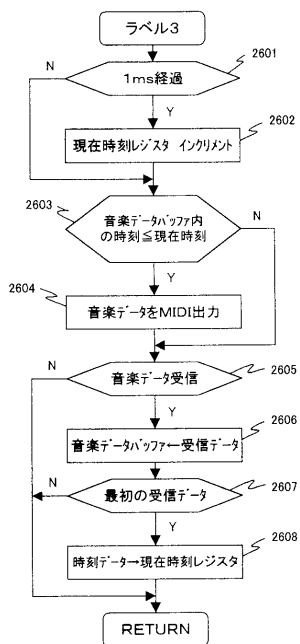
【 図 2 5 】

受信装置で実行される音楽情報  
受信処理の動作フローチャート  
(第3の実施の形態)



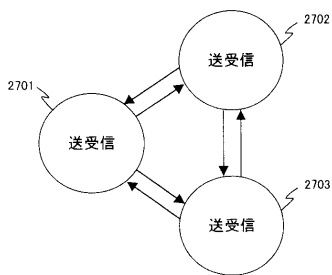
【 図 2 6 】

ラベル3処理の動作フローチャート  
(第3の実施の形態)



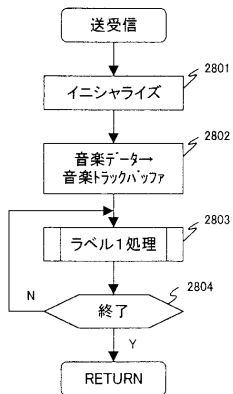
【 図 2 7 】

第4の実施の形態による音楽情報送  
受信装置を搭載した楽音発生装置の  
利用方法を説明する図



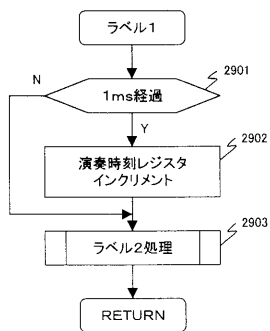
【図28】

音楽情報送受信処理の動作フローチャート  
(第4の実施の形態)



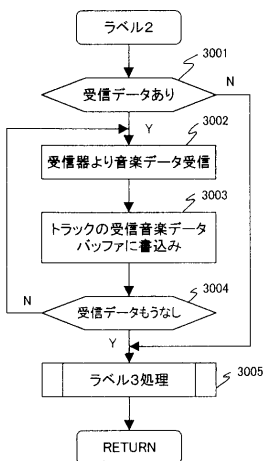
【図29】

ラベル1処理の動作フローチャート  
(第4の実施の形態)



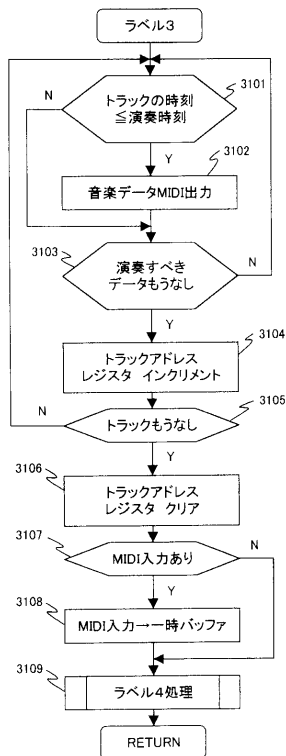
【図30】

ラベル2処理の動作フローチャート  
(第4の実施の形態)

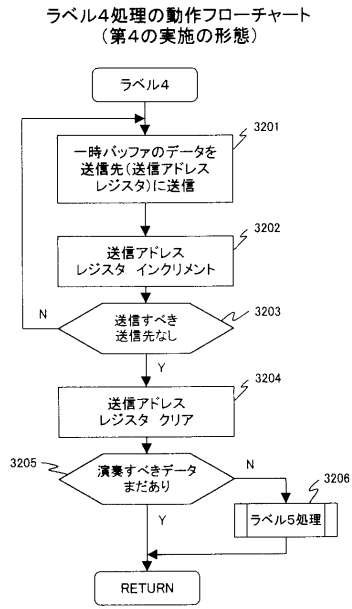


【図31】

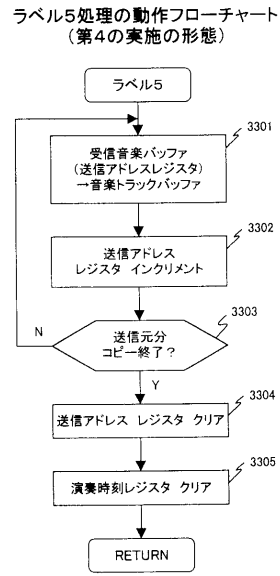
ラベル3処理の動作フローチャート  
(第4の実施の形態)



【 図 3 2 】



【 図 3 3 】



---

フロントページの続き

- (56)参考文献 特開平07 - 121161 (JP, A)  
特開平08 - 054888 (JP, A)  
特開平05 - 188947 (JP, A)  
特開平02 - 204794 (JP, A)  
特開平10 - 254801 (JP, A)

- (58)調査した分野(Int.Cl.<sup>7</sup>, DB名)  
G10H 1/00 - 7/12