(19)日本国特許庁(JP)　(12)特 許 公 報(B2)　(11)特許番号
特許第6883613号
(P6883613)

(45)発行日 令和3年6月9日(2021.6.9)　(24)登録日 令和3年5月12日(2021.5.12)

(51)Int.Cl.　F I
A63F 7/02 (2006.01)　Ａ６３Ｆ 7/02 ３２６Ｚ

請求項の数 1 (全 95 頁)

(21)出願番号 特願2019-114436(P2019-114436)
(22)出願日 令和1年6月20日(2019.6.20)
(62)分割の表示 特願2017-96738(P2017-96738)の分割
原出願日 平成29年5月15日(2017.5.15)
(65)公開番号 特開2019-171143(P2019-171143A)
(43)公開日 令和1年10月10日(2019.10.10)
審査請求日 令和2年3月17日(2020.3.17)

(73)特許権者 391010943
株式会社藤商事
大阪府大阪市中央区内本町一丁目１番４号
(74)代理人 100116942
弁理士 岩田 雅信
(74)代理人 100167704
弁理士 中川 裕人
(72)発明者 神崎 純一
大阪府大阪市中央区内本町一丁目１番４号
株式会社藤商事内
(72)発明者 柴田 伸美
大阪府大阪市中央区内本町一丁目１番４号
株式会社藤商事内

審査官 荒井 隆一

最終頁に続く

(54)【発明の名称】遊技機

(57)【特許請求の範囲】

【請求項１】

遊技動作の進行制御を行う主制御手段と、

前記主制御手段からの指示情報に基づき演出制御を行う演出制御手段と、を備え、

前記演出制御手段は、

画像表示以外の演出の実行を管理する第一手段と、

画像表示に関する演出の実行を管理する第二手段と、

前記第二手段による指示に基づき、前記画像表示に関する描画処理を実行する描画手段と、

前記主制御手段からの前記指示情報を記憶する指示情報記憶手段と、

前記指示情報記憶手段に記憶された前記指示情報を解析する解析手段と、

前記解析手段の解析結果によって得られた前記第二手段についての指示情報である第二指示情報を記憶する第二指示情報記憶手段と、を有し、

前記第一手段と前記第二手段が、共通のプロセッサによる異なるプログラムモジュールの処理で実現され、

前記解析手段による処理を、前記描画手段による前記描画処理と並行的に実行可能に構成した

遊技機。

【発明の詳細な説明】

【技術分野】

【０００１】
本発明は弾球遊技機、回胴遊技機などの遊技機に係り、特には遊技機における演出制御に関する。
【先行技術文献】
【特許文献】
【０００２】
【特許文献１】特開２０１６－１４０５９７号公報
【背景技術】
【０００３】
弾球遊技機（いわゆるパチンコ遊技機）や回胴遊技機（いわゆるスロット遊技機）等では、演出映像等の各種画像を表示する液晶表示装置等による画像表示部、ＬＥＤ（Light Emitting Diode）等を用いた演出のための点灯、点滅を行う発光部、音楽や効果音、セリフ等の音による演出を行う音出力部、モータによって駆動される可動役物等、演出動作を実行する演出手段が搭載されている。これらの演出手段はそれぞれ連携して制御されることで、興趣を向上させる遊技演出が実現される。
【０００４】
遊技機における演出動作は多様化しており、演出制御に係る処理負担も増大している。そのため、遊技動作の進行制御を行うコンピュータ装置（主制御手段）とは別に、演出動作の制御を行うためのコンピュータ装置（演出制御手段）を設けた構成が一般化している。特に、画像表示に係る制御については比較的処理負担が大きいため、演出制御用のコンピュータ装置としては、光や音等の画像演出以外の演出動作について制御を行うコンピュータ装置とは別途に、画像演出について制御を行うコンピュータ装置を実装するタイプの遊技機が存在している（例えば上記特許文献１を参照）。
【発明の概要】
【発明が解決しようとする課題】
【０００５】
ただし、演出制御用のコンピュータ装置を複数設けることはコスト面等から望ましくなく、コンピュータ装置の処理能力向上の背景もあり、画像演出以外の演出制御と画像演出の制御の双方を一つのコンピュータ装置が行うようにするいわゆる１ＣＰＵ（Central Processing Unit）化の流れがある。
１ＣＰＵ化を実現するにあたっては、画像演出以外の演出制御用のプログラムと画像演出制御用のプログラムとを統合した新たなプログラムを作成し、該プログラムを１ＣＰＵとしてのコンピュータ装置に実行させることが考えられる。
しかしながら、別々であったプログラムを１つのプログラムに統合するにあたっては元のプログラムを書き替えることを要し、開発コストの増大化を招く虞がある。
【０００６】
そこで本発明では、演出制御手段を１ＣＰＵで実現するにあたってのコスト削減を図ることを目的とする。
【課題を解決するための手段】
【０００７】
本発明に係る遊技機は、遊技動作の進行制御を行う主制御手段と、前記主制御手段からの指示情報に基づき演出制御を行う演出制御手段と、を備え、前記演出制御手段は、画像表示以外の演出の実行を管理する第一手段と、画像表示に関する演出の実行を管理する第二手段と、前記第二手段による指示に基づき、前記画像表示に関する描画処理を実行する描画手段と、前記主制御手段からの前記指示情報を記憶する指示情報記憶手段と、前記指示情報記憶手段に記憶された前記指示情報を解析する解析手段と、前記解析手段の解析結果によって得られた前記第二手段についての指示情報である第二指示情報を記憶する第二指示情報記憶手段と、を有し、前記第一手段と前記第二手段が、共通のプロセッサによる異なるプログラムモジュールの処理で実現され、前記解析手段による処理を、前記描画手段による前記描画処理と並行的に実行可能に構成したものである。

【０００８】

　上記のように演出制御手段が第二指示情報記憶手段を有していることで、当該演出制御手段をいわゆる１ＣＰＵとして構成する場合には、主制御手段からの指示情報を解析し、解析により得た第二指示情報を所定記憶領域にセットするという処理については、２ＣＰＵの場合における演出制御手段の処理を踏襲することが可能とされる。

　また、画像演出の制御処理についても、第二指示情報記憶手段にセットされた第二指示情報を取得し、該第二指示情報に基づいて画像表示装置を制御するという、２ＣＰＵの場合における画像演出制御手段と同様の処理を踏襲することが可能とされる。

　このように、第二指示情報記憶手段を有していることで、１ＣＰＵとしての演出制御手段が実行すべきプログラムとしては、２ＣＰＵ時の演出制御手段が実行していたプログラムと画像演出制御手段が実行していたプログラムをほぼそのまま踏襲することが可能とされる。

【発明の効果】

【００１１】

　本発明によれば、演出制御手段を１ＣＰＵで実現するにあたってのコスト削減を図ることができる。

【図面の簡単な説明】

【００１２】

【図１】本発明に係る実施形態のパチンコ遊技機の斜視図である。
【図２】実施形態のパチンコ遊技機の盤面の正面図である。
【図３】第一実施形態のパチンコ遊技機の制御構成のブロック図である。
【図４】実施形態のドライバ部の接続構成のブロック図である。
【図５】実施形態のドライバ部の内部構成例のブロック図である。
【図６】実施形態の主制御メイン処理のフローチャートである。
【図７】実施形態の主制御タイマ割込処理のフローチャートである。
【図８】実施形態の特別図柄管理処理のフローチャートである。
【図９】実施形態の第１特図始動口チェック処理のフローチャートである。
【図１０】実施形態のコマンドの説明図である。
【図１１】第一実施形態の演出制御メイン処理のフローチャートである。
【図１２】第一実施形態の演出制御の１ｍｓタイマ割込処理のフローチャートである。
【図１３】実施形態の演出制御におけるコマンド解析処理のフローチャートである。
【図１４】実施形態の演出制御におけるコマンド対応処理のフローチャートである。
【図１５】実施形態の演出制御におけるコマンド対応処理のフローチャートである。
【図１６】実施形態の演出制御におけるシナリオ登録処理のフローチャートである。
【図１７】実施形態の演出制御におけるシナリオ削除処理のフローチャートである。
【図１８】第一実施形態の演出制御におけるシナリオ更新処理のフローチャートである。
【図１９】第一実施形態のサブシナリオ更新処理のフローチャートである。
【図２０】実施形態のシナリオ登録情報、ランプデータ登録情報、モータデータ登録情報の説明図である。
【図２１】実施形態の音データ登録情報の説明図である。
【図２２】実施形態のメインシナリオテーブルの説明図である。
【図２３】実施形態のサブシナリオテーブルの説明図である。
【図２４】ＬＥＤ駆動データ更新処理のフローチャートである。
【図２５】ＬＥＤ出力処理のフローチャートである。
【図２６】実施形態のランプデータの登録情報の説明図である。
【図２７】実施形態のランプデータ及びマスクデータのアドレステーブルの説明図である。
【図２８】先行例のランプデータテーブル及びマスクデータテーブルの説明図である。
【図２９】実施形態のランプデータの更新及び出力の説明図である。
【図３０】実施形態のドライバへ送信するシリアルデータの説明図である。

【図３１】実施形態の各チャネルへのシリアルデータ送信の説明図である。
【図３２】実施形態の演出制御部のシリアルデータ送信動作の説明図である。
【図３３】実施形態の４ビット点灯データの８ビット変換の説明図である。
【図３４】実施形態のランプデータテーブルの説明図である。
【図３５】実施形態の変換テーブルの説明図である。
【図３６】実施形態のＬＥＤ駆動データ更新処理のフローチャートである。
【図３７】変形例のランプデータテーブルの説明図である
【図３８】第二実施形態のパチンコ遊技機の制御構成のブロック図である。
【図３９】第二実施形態の演出制御手段のブロック図である。
【図４０】第二実施形態の演出制御手段の表示制御系のブロック図である。
【図４１】２ＣＰＵ構成とした場合における演出制御部、液晶制御基板の各機能構成を模式的に表した機能ブロック図である。
【図４２】１ＣＰＵ構成とした第二実施形態の演出制御部の機能構成を模式的に表した機能ブロック図である。
【図４３】第二実施形態の演出制御メイン処理のフローチャートである。
【図４４】第二実施形態のフレーム終了時処理のフローチャートである。
【図４５】第二実施形態の演出制御タイマ割込処理のフローチャートである。
【図４６】表示データのフレーム期間内における各種タイミングを示した図である。
【図４７】演出制御コマンドと液晶制御コマンドとの対応関係を例示した図である。
【図４８】演出制御コマンドに対応して定められた各演出要素についてのシナリオデータの例を模式的に表した図である。
【図４９】演出制御コマンドの受信に応じて第二実施形態の演出制御部が実行する処理の流れを説明するための図である。

【発明を実施するための形態】

【００１３】

以下、本発明に係る遊技機の実施形態としてパチンコ遊技機を例に挙げ、次の順序で説明する。

＜１．第一実施形態＞
［1-1．パチンコ遊技機の構造］
［1-2．第一実施形態のパチンコ遊技機の制御構成］
［1-3．動作の概要］
－図柄変動表示ゲーム－
－遊技状態－
－当りについて－
［1-4．ドライバ構成及び提示用ＬＥＤ］
［1-5．主制御部の処理］
［1-6．演出制御部の処理］
（1-6-1：メイン処理）
（1-6-2：１ｍｓタイマ割込処理）
（1-6-3：コマンド解析処理）
（1-6-4：シナリオ登録・削除処理）
（1-6-5：シナリオ更新処理）
（1-6-6：ＬＥＤ駆動データ更新処理）
－先行例におけるランプデータテーブルの構造－
－ＬＥＤ駆動データ更新処理の続き－
（1-6-7：ＬＥＤ出力処理）
－点灯データのビット変換について－
［1-7．実施形態の発光駆動手法］
（1-7-1．実施形態の発光駆動情報格納手法）
（1-7-2．実施形態のＬＥＤ駆動データ更新処理）

（1-7-3．発光駆動情報格納手法の変形例）
［1-8．第一実施形態のまとめ及び変形例］
＜２．第二実施形態＞
［2-1．第二実施形態のパチンコ遊技機の制御構成］
［2-2．演出制御部の構成］
［2-3．演出制御部の機能対比］
［2-4．第二実施形態の演出制御部の処理］
［2-5．演出制御の動作例］
［2-6．第二実施形態のまとめ及び変形例］

【００１４】

＜１．第一実施形態＞

［1-1．パチンコ遊技機の構造］

まず図１、図２を参照して、本発明の実施形態としてのパチンコ遊技機１の構成を概略的に説明する。

図１は実施形態のパチンコ遊技機１の外観を示す正面側の斜視図であり、図２は遊技盤の正面図である。

図１，図２に示すパチンコ遊技機１は、主に「枠部」と「遊技盤部」から成る。

「枠部」は以下説明する前枠２、外枠４、ガラス扉５、操作パネル７を有して構成される。「遊技盤部」は図２の遊技盤３から成る。以下の説明上で、「枠部」又は「枠側」とは前枠２，外枠４、ガラス扉５、操作パネル７の総称とする。また「盤部」又は「盤側」とは遊技盤３を示す。

【００１５】

図１に示すようにパチンコ遊技機１は、木製の外枠４の前面に額縁状の前枠２が開閉可能に取り付けられている。図示していないが、この前枠２の裏面には遊技盤収納フレームが形成されており、その遊技盤収納フレーム内に図２に示す遊技盤３が装着される。これにより遊技盤３の表面に形成した遊技領域３ａが前枠２の開口部２ａから図１の遊技機前面側に臨む状態となる。

なお遊技領域３ａの前側には、透明ガラスを支持したガラス扉５が設けられており、遊技領域３ａは透明ガラスを介して前面の遊技者側に表出される。

【００１６】

ガラス扉５は軸支機構６により前枠２に対して開閉可能に取り付けられている。そしてガラス扉５の所定位置に設けられた図示しない扉ロック解除用キーシリンダを操作することで、前枠２に対するガラス扉５のロック状態を解除し、ガラス扉５を前側に開放できる構造とされている。また扉ロック解除用キーシリンダの操作によっては、外枠４に対する前枠２のロック状態も解除可能な構成とされている。

またガラス扉５の前面側には、枠側の発光手段として発光部２０ｗが各所に設けられている。発光部２０ｗは、例えばＬＥＤによる発光動作として、演出用の発光動作、エラー告知用の発光動作、動作状態に応じた発光動作などを行う。

【００１７】

ガラス扉５の下側には操作パネル７が設けられている。この操作パネル７も、図示しない軸支機構により、前枠２に対して開閉可能とされている。

操作パネル７には、上受け皿ユニット８、下受け皿ユニット９、発射操作ハンドル１０が設けられている。

【００１８】

上受け皿ユニット８には、弾球に供される遊技球を貯留する上受け皿８ａが形成されている。下受け皿ユニット９には、上受け皿８ａに貯留しきれない遊技球を貯留する下受け皿９ａが形成されている。

また上受け皿ユニット８には、上受け皿８ａに貯留された遊技球を下受け皿９ａ側に抜くための球抜きボタン１６が設けられている。下受け皿ユニット９には、下受け皿９ａに

貯留された遊技球を遊技機下方に抜くための球抜きレバー１７が設けられている。
また上受け皿ユニット８には、図示しない遊技球貸出装置に対して遊技球の払い出しを要求するための球貸しボタン１４と、遊技球貸出装置に挿入した有価価値媒体の返却を要求するためのカード返却ボタン１５とが設けられている。
さらに上受け皿ユニット８には、演出ボタン１１，１２、十字キー１３が設けられている。演出ボタン１１，１２は、所定の入力受付期間中に内蔵ランプが点灯されて操作可能となり、その内蔵ランプ点灯時に押下することにより演出に変化をもたらすことができる押しボタンとされる。また十字キー１３は遊技者が演出状況に応じた操作や演出設定等のための操作を行う操作子である。
【００１９】
発射操作ハンドル１０は操作パネル７の右端部側に設けられ、遊技者が弾球のために図３に示す発射装置５６を作動させる操作子である。
また前枠２の上部の両側と、発射操作ハンドル１０の近傍には、演出音を音響出力するスピーカ２５が設けられている。
【００２０】
次に図２を参照して、遊技盤３の構成について説明する。遊技盤３は、略正方形状の木製合板または樹脂板を主体として構成されている。この遊技盤３には、発射された遊技球を案内する球誘導レール３１が盤面区画部材として環状に装着されており、この球誘導レール３１に取り囲まれた略円形状の領域が遊技領域３ａとなっている。
【００２１】
この遊技領域３ａの略中央部には、液晶表示装置（ＬＣＤ：Liquid Crystal Display）３２が設けられている。
液晶表示装置３２では、後述する演出制御部５１の制御の下、背景画像上で、たとえば左、中、右の３つの装飾図柄の変動表示が行われる。また通常演出、リーチ演出、スーパーリーチ演出などの各種の演出画像の表示も行われる。
【００２２】
また遊技領域３ａ内には、液晶表示装置３２の表示面の周囲を囲むように、センター飾り３５Ｃが設けられている。
センター飾り３５Ｃは、そのデザインにより装飾効果を発揮するだけでなく、周囲の遊技球から液晶表示装置３２の表示面を保護する作用を持つ。さらにセンター飾り３５Ｃは、遊技球の打ち出しの強さまたはストローク長による遊技球の流路の左右打ち分けを可能とする部材としても機能する。即ち球誘導レール３１を介して遊技領域３ａ上部に打ち出された遊技球の流下経路は、センター飾り３５Ｃによって分割された左遊技領域３ｂと右遊技領域３ｃのいずれかを流下することとなる。いわゆる左打ちの場合、遊技球は左遊技領域３ｂを流下していき、右打ちの場合、遊技球は右遊技領域３ｃを流下していく。
【００２３】
また左遊技領域３ｂの下方には、左下飾り３５Ｌが設けられ、装飾効果を発揮するとともに左遊技領域３ｂとしての範囲を規定する。
同様に右遊技領域３ｃの下方には右下飾り３５Ｒが設けられ、装飾効果を発揮するとともに左遊技領域３ｂとしての範囲を規定する。
なお、遊技領域３ａ（左遊技領域３ｂ及び右遊技領域３ｃ）内には、所要各所に釘４９や風車４７が設けられて遊技球の多様な流下経路を形成する。
また液晶表示装置３２の下方にはセンターステージ３５Ｓが設けられており、装飾効果を発揮するとともに、遊技球の遊動領域として機能する。
なお図示していないが、センター飾り３５Ｃには、適所に視覚的演出効果を奏する可動体役物が設けられている。
【００２４】
遊技領域３ａの右上縁付近には、複数個のＬＥＤを配置して形成されたドット表示器による図柄表示部３３が設けられている。
この図柄表示部３３では、所定のドット領域により、第１特別図柄表示部、第２特別図

柄表示部、及び普通図柄表示部が形成され、第１特別図柄、第２特別図柄、及び普通図柄のそれぞれの変動表示動作（変動開始および変動停止を一セットする変動表示動作）が行われる。
なお、液晶表示装置３２は、図柄表示部３３による第１、第２特別図柄の変動表示と時間的に同調して、画像による装飾図柄を変動表示する。
【００２５】
センター飾り３５Ｃの下方には、上始動口４１（第１の特別図柄始動口）を有する入賞装置が設けられ、さらにその下方には下始動口４２ａ（第２の特別図柄始動口）を備える普通変動入賞装置４２が設けられている。
上始動口４１及び下始動口４２ａの内部には、遊技球の通過を検出する検出センサ（図３に示す上始動口センサ７１，下始動口センサ７２）が形成されている。
【００２６】
上始動口４１は、図柄表示部３３における第１特別図柄の変動表示動作の始動条件に係る入賞口で、始動口開閉手段（始動口を開放または拡大可能にする手段）を有しない入賞率固定型の入賞装置となっている。
【００２７】
下始動口４２ａを有する普通変動入賞装置４２は、始動口開閉手段により始動口の遊技球の入賞率を変動可能な入賞率変動型の入賞装置として構成されている。即ち下始動口４２ａを開放または拡大可能にする左右一対の可動翼片（可動部材）４２ｂを備えた、いわゆる電動チューリップ型の入賞装置である。
この普通変動入賞装置４２の下始動口４２ａは、図柄表示部３３における第２特別図柄の変動表示動作の始動条件に係る入賞口である。そして、この下始動口４２ａの入賞率は可動翼片４２ｂの作動状態に応じて変動する。即ち可動翼片４２ｂが開いた状態では、入賞が容易となり、可動翼片４２ｂが閉じた状態では、入賞が困難又は不可能となるように構成されている。
【００２８】
また普通変動入賞装置４２の左右には、一般入賞口４３が複数個設けられている。各一般入賞口４２の内部には、遊技球の通過を検出する検出センサ（図３に示す一般入賞口センサ７４）が形成されている。
また右遊技領域３ｃの下部側には、遊技球が通過可能なゲート（特定通過領域）からなる普通図柄始動口４４が設けられている。この普通図柄始動口４４は、図柄表示部３３における普通図柄の変動表示動作に係る入賞口であり、その内部には、通過する遊技球を検出するセンサ（図３に示すゲートセンサ７３）が形成されている。
【００２９】
右遊技領域３ｃ内の普通図柄始動口４４から普通変動入賞装置４２へかけての流下経路途中には第１特別変動入賞装置４５（特別電動役物）が設けられている。
第１特別変動入賞装置４５は、突没式の開放扉４５ｂにより第１大入賞口４５ａを閉鎖／開放する構造とされている。また、その内部には第１大入賞口４５ａへの遊技球の通過を検出するセンサ（図３の第１大入賞口センサ７５）が形成されている。
第１大入賞口４５ａの周囲は、右下飾り３５Ｒが遊技盤３の表面から膨出した状態となっており、その膨出部分の上辺及び開放扉４５ｂの上面が右流下経路３ｃの下流案内部を形成している。従って、開放扉４５ｂが盤内部側に引き込まれることで、下流案内部に達した遊技球は容易に第１大入賞口４５ａに入る状態となる。
【００３０】
また普通変動入賞装置４２の下方には、第２特別変動入賞装置４６（特別電動役物）が設けられている。第２特別変動入賞装置４６は、下部が軸支されて開閉可能な開放扉４６ｂにより、その内側の第２大入賞口４６ａを閉鎖／開放する構造とされている。また、その内部には第２大入賞口４６ａへの遊技球の通過を検出するセンサ（図３の第２大入賞口センサ７６）が形成されている。
開放扉４６ｂが開かれることで第２大入賞口４６ａが開放される。この状態では、左遊

技領域３ｂ或いは右遊技領域３ｃを流下してきた遊技球は、高い確率で第２大入賞口５０に入ることとなる。

【００３１】

以上のように盤面の遊技領域には、入賞口として上始動口４１、下始動口４２ａ、普通図柄始動口４４、第１大入賞口４５ａ、第２大入賞口４６ａ、一般入賞口４３が形成されている。

本実施形態のパチンコ遊技機１においては、これら入賞口のうち、普通図柄始動口４４以外の入賞口への入賞があった場合には、各入賞口別に設定された入賞球１個当りの賞球数が遊技球払出装置５５（図３参照）から払い出される。

例えば、上始動口４１および下始動口４２ａは３個、第１大入賞口４５ａ、第２大入賞口４６ａは１３個、一般入賞口４３は１０個などと賞球数が設定されている。

なお、これらの各入賞口に入賞しなかった遊技球は、アウト口４８を介して遊技領域３ａから排出される。

ここで「入賞」とは、入賞口がその内部に遊技球を取り込んだり、ゲートを遊技球が通過したりすることをいう。実際には入賞口ごとに形成されたセンサ（各入賞検出スイッチ）により遊技球が検出された場合、その入賞口に「入賞」が発生したものとして扱われる。この入賞に係る遊技球を「入賞球」とも称する。

【００３２】

以上のような盤面において、センター飾り３５Ｃ、左下飾り３５Ｌ、右下飾り３５Ｒ、センターステージ３５Ｓ、第１特別変動入賞装置４５、第２特別変動入賞装置４６、さらには図示していない可動体役物には、詳細には図示していないが各所に、盤側の発光手段として例えばＬＥＤによる発光部２０ｂが設けられている。

発光部２０ｂは演出用の発光動作やエラー告知用の発光動作、動作状態に応じた発光動作などを行う。

【００３３】

以下では説明上、盤側の発光部２０ｗ、枠側の発光部２０ｂを「演出用発光部」と表記する場合がある。

【００３４】

［1-2. 第一実施形態のパチンコ遊技機の制御構成］

次に本実施形態のパチンコ遊技機１の制御系の構成について説明する。図３はパチンコ遊技機１の内部構成の概略的なブロック図である。

本実施形態のパチンコ遊技機１は、その制御構成を形成する基板として主に、主制御基板５０、演出制御基板５１、液晶制御基板５２、払出制御基板５３、発射制御基板５４、電源基板５８が設けられている。

【００３５】

主制御基板５０は、マイクロコンピュータ等が搭載され、パチンコ遊技機１の遊技動作全般に係る統括的な制御を行う。なお以下では、主制御基板５０に搭載されたマイクロコンピュータ等を含めて主制御基板５０の構成体を「主制御部５０」と表記する。

演出制御基板５１は、マイクロコンピュータ等が搭載され、主制御部５０から演出制御コマンドを受けて、画像表示、発光、音響出力、可動体役物を用いた各種の演出動作を実行させるための制御を行う。なお以下では、演出制御基板５１に搭載されたマイクロコンピュータ等を含めて演出制御基板５１の構成体を「演出制御部５１」と表記する。

【００３６】

液晶制御基板５２はマイクロコンピュータやビデオプロセッサ等が搭載され、演出制御部５１からの表示制御コマンドを受けて、液晶表示装置３２による表示動作の制御を行う。

払出制御基板５３は、パチンコ遊技機１に接続された遊技球払出装置５５による賞球の払い出し制御を行う。

発射制御基板５４は、遊技者のパチンコ遊技機１に設けられている発射装置５６による

遊技球の発射動作の制御を行う。

電源基板５８は、外部電源（例えばＡＣ２４Ｖ）からＡＣ／ＤＣ変換、さらにはＤＣ／ＤＣ変換を行い、各部に動作電源電圧Ｖｃｃを供給する。なお電源経路の図示は省略している。

【００３７】

まず主制御部５０及びその周辺回路について述べる。

主制御部５０は、ＣＰＵ１００（以下「主制御ＣＰＵ１００」と表記）を内蔵したマイクロプロセッサ、ＲＯＭ１０１（以下「主制御ＲＯＭ１０１」と表記）、ＲＡＭ１０２（以下「主制御ＲＡＭ１０２」と表記）を搭載し、マイクロコンピュータを構成している。

主制御ＣＰＵ１００は制御プログラムに基づいて、遊技の進行に応じた各種演算及び制御処理を実行する。

主制御ＲＯＭ１０１は、主制御ＣＰＵ１００による遊技動作の制御プログラムや、遊技動作制御に必要な種々のデータを記憶する。

主制御ＲＡＭ１０２は、主制御ＣＰＵ１００が各種演算処理に使用するワークエリアや、各種入出力データや処理データのバッファ領域として用いられる。

なお図示は省略したが、主制御部５０は、各部とのインターフェース回路、特別図柄変動表示に係る抽選用乱数を生成する乱数生成回路、各種の時間計数のためのＣＴＣ（Counter Timer Circuit）、主制御ＣＰＵ１００に割込み信号を与える割込コントローラ回路なども備えている。

【００３８】

主制御部５０は、上述のように盤面の遊技領域の各入賞手段（上始動口４１、下始動口４２ａ、普通図柄始動口４４、第１大入賞口４５ａ、第２大入賞口４６ａ、一般入賞口４３）に設けられるセンサの検出信号を受信する構成となっている。

即ち、上始動口センサ７１、下始動口センサ７２、ゲートセンサ７３、一般入賞口センサ７４、第１大入賞口センサ７５、第２大入賞口センサ７６のそれぞれの検出信号が主制御部５０に供給される。

なお、これらのセンサ（７１～７６）は、入球した遊技球を検出する検出スイッチにより構成されるが、具体的にはフォトスイッチや近接スイッチなどの無接点スイッチや、マイクロスイッチなどの有接点スイッチで構成することができる。

【００３９】

主制御部５０は、上始動口センサ７１、下始動口センサ７２、ゲートセンサ７３、一般入賞口センサ７４、第１大入賞口センサ７５、第２大入賞口センサ７６のそれぞれの検出信号の受信に応じて、処理を行う。例えば抽選処理、図柄変動制御、賞球払出制御、演出制御コマンド送信制御、外部データ送信処理などを行う。

【００４０】

また主制御部５０には、下始動口４２の可動翼片４２ｂを開閉駆動する普通電動役物ソレノイド７７が接続され、主制御部５０は遊技進行状況に応じて制御信号を送信して普通電動役物ソレノイド７７の駆動動作を実行させ、可動翼片４２ｂの開閉動作を実行させる。

さらに、主制御部５０には、第１大入賞口４５の開放扉４５ｂを開閉駆動する第１大入賞口ソレノイド７８と、第２大入賞口４６の開放扉４６ｂを開閉駆動する第２大入賞口ソレノイド７９が接続されている。主制御部５０は、いわゆる大当たり状況に応じて、第１大入賞口ソレノイド７８又は第２大入賞口ソレノイド７９を駆動制御して、第１大入賞口４５又は第２大入賞口４６の開放動作を実行させる。

【００４１】

また主制御部５０には、図柄表示部３３が接続されており、図柄表示部３３に制御信号を送信して、各種図柄表示（ＬＥＤの消灯／点灯／点滅）を実行させる。これにより図柄表示部３３における第１特別図柄表示部８０、第２特別図柄表示部８１、普通図柄表示部８２での表示動作が実行される。

【００４２】

また主制御部５０には、枠用外部端子基板５７が接続される。主制御部５０は、遊技進行に関する情報を、枠用外部端子基板５７を介して図示しないホールコンピュータに送信可能となっている。遊技進行に関する情報とは、例えば大当り当選情報、賞球数情報、図柄変動表示実行回数情報などの情報である。ホールコンピュータとは、パチンコホールの遊技機を統括的に管理する管理コンピュータであり、遊技機外部に設置されている。

【００４３】

また主制御部５０には、払出制御基板５３が接続されている。払出制御基板５３には、発射装置５６を制御する発射制御基板５４と、遊技球の払い出しを行う遊技球払出装置５５が接続されている。

主制御部５０は、払出制御基板５３に対し、払い出しに関する制御コマンド（賞球数を指定する払出制御コマンド）を送信する。払出制御基板５３は当該制御コマンドに応じて遊技球払出装置５５を制御し、遊技球の払い出しを実行させる。

また払出制御基板５３は、主制御部５０に対して、払い出し動作状態に関する情報（払出状態信号）を送信可能となっている。主制御部５０側では、この払出状態信号によって、遊技球払出装置５５が正常に機能しているか否かを監視する。具体的には、賞球の払い出し動作の際に、玉詰まりや賞球の払い出し不足といった不具合が発生したか否かを監視している。

【００４４】

また主制御部５０は、特別図柄変動表示に関する情報を含む演出制御コマンドを、演出制御部５１に送信する。なお、主制御部５０から演出制御部５１への演出制御コマンドの送信は一方向通信により実行されるようにしている。これは、外部からの不正行為による不正な信号が演出制御部５１を介して主制御部５０に入力されることを防止するためである。

【００４５】

続いて演出制御部５１及びその周辺回路について説明する。

演出制御部５１は、ＣＰＵ２００（以下「演出制御ＣＰＵ２００」と表記）を内蔵したマイクロプロセッサ、ＲＯＭ２０１（以下「演出制御ＲＯＭ２０１」と表記）、ＲＡＭ２０２（以下「演出制御ＲＡＭ２０２」と表記）を搭載し、マイクロコンピュータを構成している。

演出制御ＣＰＵ２００は演出制御プログラム及び主制御部５０から受信した演出制御コマンドに基づいて、各種演出動作のための演算処理や各演出手段の制御を行う。演出手段とは、本実施形態のパチンコ遊技機１の場合、液晶表示装置３２、発光部２０ｗ、２０ｂ、スピーカ２５、及び図示を省略した可動体役物となる。

また演出制御ＣＰＵ２００は各種の状況提示のための制御も行う。

演出制御ＲＯＭ２０１は、演出制御ＣＰＵ２００による演出動作の制御プログラムや、演出動作制御に必要な種々のデータを記憶する。

演出制御ＲＡＭ２０２は、演出制御ＣＰＵ２００が各種演算処理に使用するワークエリアや、テーブルデータ領域、各種入出力データや処理データのバッファ領域などとして用いられる。

なお図示は省略したが、演出制御部５１は、各部とのインターフェース回路、演出のための抽選用乱数を生成する乱数生成回路、各種の時間計数のためのＣＴＣ、演出制御ＣＰＵ２００に割込み信号を与える割込コントローラ回路、音響演出のための音源ＩＣなども備えている。

この演出制御部５１の主な役割は、主制御部５０からの演出制御コマンドの受信、演出制御コマンドに基づく演出の選択決定、液晶制御基板５２側への液晶制御コマンドの送信、スピーカ４５の音声出力制御、発光部２０ｗ、２０ｂ（ＬＥＤ）の発光制御、可動体役物の動作制御などとなる。

【００４６】

演出制御部５１は、液晶制御基板５２側への液晶制御コマンドの送信を行うが、その液晶制御コマンドは、液晶インターフェース基板６６を介して液晶制御基板５２に送られる

。

【００４７】

液晶制御基板５２は、液晶表示装置３２の表示制御を行う。図示していないが、液晶制御基板５２には、ＶＤＰ（Video Display Processor）、画像ＲＯＭ、ＶＲＡＭ（Video RAM）、液晶制御ＣＰＵ、液晶制御ＲＯＭ、液晶制御ＲＡＭを備えている。

ＶＤＰは、画像展開処理や画像の描画などの映像出力処理全般の制御を行う。

画像ＲＯＭには、ＶＤＰが画像展開処理を行う画像データ（演出画像データ）が格納されている。

ＶＲＡＭは、ＶＤＰが展開した画像データを一時的に記憶する画像メモリ領域とされる。

液晶制御ＣＰＵは、ＶＤＰが表示制御を行うために必要な制御データを出力する。

液晶制御ＲＯＭには、液晶制御ＣＰＵの表示制御動作手順を記述したプログラムやその表示制御に必要な種々のデータが格納される。

液晶制御ＲＡＭは、ワークエリアやバッファメモリとして機能する。

【００４８】

液晶制御基板５２は、これらの構成により、演出制御基板５１からの液晶制御コマンドに基づいて各種の画像データを生成し、液晶表示装置３２に出力する。これによって液晶表示装置３２において各種の演出画像が表示される。

具体的に、液晶制御基板５２の液晶制御ＣＰＵは、演出制御部５１が主制御部５０からの受信コマンドを解析して得た液晶制御コマンドを取得し、該液晶制御コマンドについての解析を行い、解析結果に基づき液晶制御用のシナリオデータ（演出進行データ）の作成、及び該シナリオデータに基づく画像データの生成を行い、液晶表示装置３２に出力する。

【００４９】

また演出制御部５１には枠ドライバ部６１、盤ドライバ部６２が接続されている。

枠ドライバ部６１は、枠側のランプ部６３のＬＥＤについて発光駆動を行う。なお、ランプ部６３とは、図１に示したように枠側に設けられている発光部２０ｗを総括的に示したものである。

盤ドライバ部６２は、盤側のランプ部６４のＬＥＤについて発光駆動を行う。なお、ランプ部６４とは、図２に示したように盤側に設けられている発光部２０ｂを総括的に示したものである。

【００５０】

また演出制御部５１にはモータドライバ部７０が接続されている。

モータドライバ部７０は可動体役物モータ６５を駆動して、盤側に設けられている可動体役物の動作を実行させる。

可動体役物モータ６５は例えばステッピングモータが用いられる。

【００５１】

本実施形態の場合、枠ドライバ部６１は第１系統の駆動信号出力手段、盤ドライバ部６２は第２系統の駆動信号出力手段、モータドライバ部７０は第３系統の駆動信号出力手段である。

詳しくは図４を用いて後述するが、本実施形態の場合、演出制御部５１（演出制御ＣＰＵ２００）は、シリアルデータ送信チャネルｃｈ１，ｃｈ２，ｃｈ３を用いて、発光駆動データやモータ駆動データをシリアルデータとして、枠ドライバ部６１、盤ドライバ部６２、モータドライバ部７０に出力する。

なお例えば盤ドライバ部６２の一部が、モータドライバ部７０としての動作を行うことで、ランプ部６４の各ＬＥＤと可動体役物モータ６５の駆動を行うようにしてもよい。

【００５２】

また演出制御部５１は、スピーカ２５により所望の音を出力させるべく、音源ＩＣ５９に対する制御を行う。音源ＩＣ５９には音データＲＯＭ６９が接続されており、音源ＩＣ５９は音データＲＯＭ６９から必要な音データ（再生するフレーズの音データ）を取得し

て音声信号出力を行う。

音源ＩＣ５９は、複数チャネルのフレーズをミキシングして所定本数（チャネル数）の音声信号を得る。図１に示したように、本例の場合、スピーカ２５は複数設けられるため、音源ＩＣ５９の出力チャネル数は例えばＬｃｈ，Ｒｃｈの２チャネルなど（ステレオ出力）が可能となる。上記のミキシングにより、演出制御部５１より再生指示された複数チャネルのフレーズを同時再生可能とされる。

【００５３】

音源ＩＣ５９による出力音声信号はアンプ部６７で増幅された後、スピーカ２５に対して与えられる。

なお、図３では図示の都合上、音源ＩＣ５９の出力チャネル数を１つとしているが、実際にはアンプ部６７及びスピーカ２５としては、例えばＬｃｈ、Ｒｃｈに対応した出力チャネルがそれぞれ設けられ、ステレオによる音再生が可能とされる。

また、この例では音源ＩＣ５９を演出制御部５１とは別体に設けるものとしたが、音源ＩＣ５９は演出制御部５１内に設けることもできる。

【００５４】

また演出制御部５１には、遊技者が操作可能な操作部６０が接続され、操作部６０からの操作検出信号を受信可能となっている。この操作部６０は、図１で説明した演出ボタン１１，１２、十字キー１３と、それらの操作検出機構のことである。

演出制御部５１は、操作部６０からの操作検出信号に応じて、各種演出制御を行うことができる。

【００５５】

演出制御部５１は、主制御部５０から送られてくる演出制御コマンドに基づき、あらかじめ用意された複数種類の演出パターンの中から抽選によりあるいは一意に演出パターンを決定し、必要なタイミングで各種演出手段を制御する。これにより、演出パターンに対応する液晶表示装置３２による演出画像の表示、スピーカ２５からの音の再生、ランプ部６３、６４（発光部２０ｗ、２０ｂ）におけるＬＥＤの点灯点滅駆動、可動体役物モータ６５による可動体役物の動作が実現され、時系列的に種々の演出パターンが展開されていく。これにより「演出シナリオ」が実現される。

【００５６】

なお演出制御コマンドは、１バイト長のモード（ＭＯＤＥ）と、同じく１バイト長のイベント（ＥＶＥＮＴ）からなる２バイト構成により機能を定義する。

ＭＯＤＥとＥＶＥＮＴの区別を行うために、ＭＯＤＥのＢｉｔ７はＯＮ、ＥＶＥＮＴのＢｉｔ７をＯＦＦとしている。

これらの情報を有効なものとして送信する場合、モード（ＭＯＤＥ）およびイベント（ＥＶＥＮＴ）各々に対応してストローブ信号が出力される。すなわち、主制御ＣＰＵ１００は、送信すべきコマンドがある場合、演出制御部５１にコマンドを送信するためのモード（ＭＯＤＥ）情報の設定および出力を行い、この設定から所定時間経過後に１回目のストローブ信号の送信を行う。さらに、このストローブ信号の送信から所定時間経過後にイベント（ＥＶＥＮＴ）情報の設定および出力を行い、この設定から所定時間経過後に２回目のストローブ信号の送信を行う。

ストローブ信号は主制御ＣＰＵ１００により、演出制御ＣＰＵ２００が確実にコマンドを受信することが可能な所定期間アクティブ状態に制御される。

また演出制御部５１（演出制御ＣＰＵ２００）は、ストローブ信号の入力に基づいて割込を発生させてコマンド受信割込処理用の制御プログラムを実行し、この割込処理において演出制御コマンドが取得される。

【００５７】

［1-3. 動作の概要］

パチンコ遊技機１の動作の概説について説明する。

－図柄変動表示ゲーム－

パチンコ遊技機１では、所定の始動条件、具体的には、遊技球が上始動口４１または下始動口４２に入賞したことを条件に、主制御部５０において乱数抽選による大当り抽選が行なわれる。この抽選結果に基づき、特別図柄（大当り抽選結果を報知するための識別図柄）を第１特別図柄表示部８０または第２特別図柄表示部８１に変動表示させて特別図柄変動表示ゲームを開始し、一定時間経過後に、その結果を特別図柄表示部（８０又は８１）に表示する。

【００５８】

本実施形態では、上始動口４１への入賞に基づく大当り抽選と下始動口４２への入賞に基づく大当り抽選とは、独立して行われる。このため、上始動口４１に関する大当り抽選結果は第１特別図柄表示部８０側、下始動口４２に関する大当り抽選結果は第２特別図柄表示部８１側で導出表示されるようになっている。

なお説明上、第１特別図柄表示部８０側の特別図柄変動表示ゲームを「第１特別図柄変動表示ゲーム」、第２特別図柄表示部８１側の特別図柄変動表示ゲームを「第２特別図柄変動表示ゲーム」と称する。但し、第１，第２特別図柄を総称して「特別図柄」と表記し、また第１，第２特別図柄変動表示ゲームを「特別図柄変動表示ゲーム」と総称する。

【００５９】

特別図柄変動表示ゲームが開始されると、これに伴い、装飾図柄（遊技図柄）を液晶表示装置３２に変動表示させる装飾図柄変動表示ゲームが開始され、これに付随して種々の演出が現出される。そして、第１，第２特別図柄表示部（８０、８１）に抽選結果が表示されると、液晶表示装置３２にも装飾図柄によりその結果が表示される。すなわち、この装飾図柄変動表示ゲームでは、特別図柄変動表示ゲームでの抽選結果を反映させた演出表示、つまり大当り抽選結果を反映させた演出が現出される。

【００６０】

例えば特別図柄変動表示ゲームの結果が「大当り」である場合、装飾図柄変動表示ゲームの結果も「大当り」を反映させた演出となる。また、特別図柄表示部（８０、８１）には、大当りを示す特別図柄が所定の表示態様で停止表示され、液晶表示装置３２には、「左」「中」「右」の各表示エリアにおいて、当り有効ライン上で装飾図柄が大当り抽選結果を反映させた所定の表示態様（例えば「左」「中」「右」の各表示エリアにおいて、３個の装飾図柄が「７」「７」「７」の表示状態）で停止表示される。

【００６１】

この「大当り」となった場合、例えば第１大入賞口ソレノイド７８又は第２大入賞口ソレノイド７９が駆動され、第１大入賞口４５又は第２大入賞口４６の所定パターンでの開放動作が実行され、通常遊技状態よりも遊技者に有利な特別遊技状態（大当り遊技）が発生する。この大当り遊技では、大入賞口の開放時間が所定時間経過するまでか、または大入賞口に所定個数の遊技球が入賞するまで開放され、いずれかを満たしたことを条件に大入賞口が所定時間閉鎖される、といったラウンド遊技があらかじめ定められた規定回数、繰り返される。

大当り遊技が開始されると、最初に大当りが開始された旨を報知するオープニング演出が行われ、オープニング演出が終了した後、上記ラウンド遊技が規定ラウンド数、行われる。また、ラウンド遊技中は、各ラウンド対応するラウンド演出が行われる。そして、規定ラウンド数終了後には、大当りが終了される旨を報知するエンディング演出が行われ、これにより大当り遊技が終了する。

【００６２】

このように、特別図柄変動表示ゲームと装飾図柄変動表示ゲームとは、その図柄遊技時間（変動表示の開始タイミングから停止表示のタイミング）とがほぼ同じとなり、特別図柄変動表示ゲームの結果を反映したものが装飾図柄変動表示ゲームにおいて表現されることとしているので、この２つの図柄変動表示ゲームを等価的な図柄遊技と捉えることもできる。説明上、上記２つの図柄変動表示ゲームを単に「図柄変動表示ゲーム」と称する場

合がある。
【００６３】
　また、遊技球がゲート４４（普通図柄始動口）を通過したことに基づき、主制御部５０において乱数抽選による補助当り抽選が行なわれる。この抽選結果に応じて普通図柄表示部８２のＬＥＤにより表現される普通図柄を変動表示させて普通図柄変動表示ゲームを開始し、所定時間経過後に、その結果をＬＥＤの点灯と非点灯の特定の組合せにて停止表示する。
【００６４】
　そして「補助当り」となった場合には、普通電動役物ソレノイド７７が作動し、可動翼片４２ｂが開いて下始動口４２が開放または拡大されて遊技球が流入し易い状態（始動口開状態）となり、通常遊技状態よりも遊技者に有利な補助遊技状態（以下、「普電開放遊技」と称する）が発生する。この普電開放遊技では、可動翼片４２ｂが所定時間（例えば０．２秒）開放されるか、または所定個数（例えば４個）の遊技球が入賞するまで開放され、その後、所定時間（例えば０．５秒）可動翼片４２ｂが閉まる、といった動作が所定回数繰り返される。なお、普電開放遊技中に遊技球が下始動口４２に入賞した場合にも、同様に上記特別図柄変動表示ゲームが行なわれ、これに伴い装飾図柄変動表示ゲームが行なわれる。
【００６５】
　ここで、特別図柄変動表示ゲーム中、普通図柄変動表示ゲーム中、大当り遊技中、または普電開放遊技などの最中に、さらに上始動口センサ７１または下始動口センサ７２もしくはゲートセンサ７３からの検出信号の入力があり、始動条件が成立した場合には、この検出信号に基づいて当り抽選に利用する遊技情報を取得し、これを、各変動表示ゲームを行わせるための始動権利に係るデータ（保留データ）として、変動表示中にかかわるものを除き、所定の上限値である最大保留記憶数（例えば最大４個）まで保留記憶可能となっている。この保留数を遊技者に明らかにするため、保留数表示部２０Ｈや液晶表示装置３２による画面中にアイコン画像として設けた保留表示器を点灯表示させる。
　通常は、この保留球の発生順に、各保留球に対する変動表示ゲームが実行される。本実施形態では、最大保留記憶個数と同数の４個を上限の所定個数として扱い、第１特別図柄、第２特別図柄に関する保留データをそれぞれ４個まで記憶し、特別図柄または普通図柄の変動確定回数として保留する。
【００６６】
　－遊技状態－

　本実施形態のパチンコ遊技機１では、複数種類の遊技状態を発生可能に構成されている。
　まず、本実施形態のパチンコ遊技機１は、主制御部５０（ＣＰＵ１００）がその機能部を担う「確率変動（以下、「確変」と称する）機能」を備えている。これには特別図柄に係る確変機能（以下「特別図柄確変機能」と称する）と普通図柄に係る確変機能（以下「普通図柄確変機能」と称する）の２種類がある。
【００６７】
　特別図柄確変機能は、大当りの抽選確率を所定確率（通常確率）の低確率（例えば３９９分の１）から高確率（例えば３９．９分の１）に変動させて、通常遊技状態よりも有利な「高確率状態」を発生させる機能である。この高確率状態下では、大当り抽選確率が高確率となることから、大当りが生起され易くなる。
　普通図柄確変機能は、補助当り抽選確率が所定確率（通常確率）である低確率（例えば２５６分の１）から高確率（例えば２５６分の２５５）に変動させて、通常遊技状態よりも有利な「補助当り確変状態」を発生させる機能である。この補助当り確変状態下では、補助当り抽選確率が高確率状態となることから補助当りが生起され易くなり、普電開放遊技が頻繁に発生して、通常遊技状態よりも単位時間当りの可動翼片４２ｂの作動率が向上する作動率向上状態となる。

【００６８】
　また、本実施形態のパチンコ遊技機１は、主制御部５０がその機能部を担う「変動時間短縮（以下「時短」と称する）機能」を備えている。これには特別図柄に係る時短機能（以下、「特別図柄時短機能」と称する）と普通図柄に係る時短機能（以下、「普通図柄時短機能」と称する）の二種類がある。
【００６９】
　特別図柄時短機能は、１回の特別図柄変動表示ゲームに要する平均時間（特別図柄が変動を開始してから確定表示される迄の時間。つまり、特別図柄の変動時間）を短縮した「特別図柄時短状態」を発生させる機能である。特別図柄時短状態下では、１回の特別図柄変動表示ゲームにおける特別図柄の平均的な変動時間が短縮され、通常遊技状態よりも単位時間当りの大当り抽選回数が向上する抽選回数向上状態となる。
　なお、パチンコ遊技機１では、特別図柄の変動表示時間が保留数の違いにより短縮される場合があるが、この場合は、特別図柄時短状態が発生しているわけではなく、他の制御処理によるものである。
【００７０】
　普通図柄時短機能は、１回の普通図柄変動表示ゲームに要する平均時間（普通図柄が変動を開始してから確定表示されるまでの時間。つまり、普通図柄の変動時間）を短縮した「普通図柄時短状態」を発生させる機能である。普通図柄時短状態下では、１回の普通図柄変動表示ゲームにおける普通図柄の平均的な変動時間が短縮され、通常遊技状態よりも単位時間当りの補助当り抽選回数が向上する抽選回数向上状態となる。
【００７１】
　また本実施形態のパチンコ遊技機１は、主制御部５０がその機能部を担う「開放延長機能」を備えている。
　開放延長機能は、可動翼片４２ｂを開動作させる期間およびその開放回数を延長した「開放延長状態」を発生させる機能である。この開放延長状態は、いわゆる「電チューサポート状態」と称される。開放延長状態下では、可動翼片４２ｂの開動作期間（始動口開状態時間）が、例えば０．２秒から１．７秒に延長され、またその開閉回数が、例えば１回から２回に延長され、通常遊技状態よりも単位時間当りの可動翼片４２ｂの作動率が向上する作動率向上状態となる。
【００７２】
　以上のような各機能を１または複数種類作動させることにより、遊技機の内部的な遊技状態に変化をもたらすことができる。
　ここで本実施形態では、普通図柄確変機能、普通図柄時短機能、および開放延長機能の作動開始条件は、特別図柄時短機能の作動開始条件と同じ条件としており、各機能が同じ契機にて動作することになる。
【００７３】
　－当りについて－

　本実施形態のパチンコ遊技機１では、特別図柄変動表示ゲームにて抽選される当りの種類、つまり大当り抽選対象となる当り種別として、「１５Ｒ低ベース非確変大当り」、「１５Ｒ低ベース確変α大当り」、「１５Ｒ低ベース確変β大当り」、「１５Ｒ高ベース確変大当り」、「２Ｒ低ベース確変大当り」、および「小当り」などの複数種類の当りが設けられている。
【００７４】
［1-4. ドライバ構成及び提示用ＬＥＤ］

　続いて、第１の実施形態としてのドライバ構成を説明する。上述した演出制御部５１のシリアルデータ送信チャネルｃｈ１，ｃｈ２，ｃｈ３により発光駆動データやモータ駆動データがシリアルデータとして供給される枠ドライバ部６１（第１系統）、盤ドライバ部６２（第２系統）、モータドライバ部７０（第３系統）の構成である。

【００７５】
　図４に演出制御部５１に接続される枠ドライバ部６１、盤ドライバ部６２、モータドライバ部７０を示した。
　第１系統の駆動信号出力手段である枠ドライバ部６１は、ｎ個のＬＥＤドライバ９０が、演出制御ＣＰＵ２００のシリアルデータ出力チャネルｃｈ１に対して並列に接続されている。
　シリアルデータ出力チャネルｃｈ１の信号線としては、リセット信号RESETを供給するリセット信号線、クロック信号CLKを供給するクロック線、発光駆動データ（「ＬＥＤ駆動データ」ともいう）としてのシリアルデータDATAを供給するデータ線、イネーブル信号ENABLEを供給するイネーブル信号線が設けられている。これら各信号線は、それぞれ、枠ドライバ部６１を構成するｎ個のＬＥＤドライバ９０に対して各信号を並列に供給するように接続されている。
　枠ドライバ部６１の各ＬＥＤドライバ９０には、演出制御ＣＰＵ２００がスレーブアドレスとして用いるデバイスＩＤが設定されている。即ち個々のＬＥＤドライバ９０の識別子である。説明上、仮に、図示のように各ＬＥＤドライバ９０のデバイスＩＤ（スレーブアドレス）をｗ１～ｗ（ｎ）と表記する。
【００７６】
　また第２系統の駆動信号出力手段である盤ドライバ部６２は、ｍ個のＬＥＤドライバ９０が、演出制御ＣＰＵ２００のシリアルデータ出力チャネルｃｈ２に対して並列に接続されている。
　シリアルデータ出力チャネルｃｈ２の信号線もチャネルｃｈ１と同様、リセット信号RESETを供給するリセット信号線、クロック信号CLKを供給するクロック線、発光駆動データとしてのシリアルデータDATAを供給するデータ線、イネーブル信号ENABLEを供給するイネーブル信号線が設けられている。これら各信号線は、それぞれ、盤ドライバ部６２を構成するｍ個のＬＥＤドライバ９０に対して各信号を並列に供給するように接続されている。
　盤ドライバ部６２の各ＬＥＤドライバ９０には、演出制御ＣＰＵ２００がスレーブアドレスとして用いるデバイスＩＤ（個々のＬＥＤドライバ９０の識別子）が設定されている。説明上、仮に、図示のように各ＬＥＤドライバ９０のデバイスＩＤ（スレーブアドレス）をｂ１～ｂ（ｍ）と表記する。
【００７７】
　また第３系統の駆動信号出力手段であるモータドライバ部７０は、ｐ個のモータドライバ９０Ｍが、演出制御ＣＰＵ２００のシリアルデータ出力チャネルｃｈ３に対して並列に接続されている。
　シリアルデータ出力チャネルｃｈ３の信号線も同様に、リセット信号RESETを供給するリセット信号線、クロック信号CLKを供給するクロック線、モータ駆動データとしてのシリアルデータDATAを供給するデータ線、イネーブル信号ENABLEを供給するイネーブル信号線が設けられている。これら各信号線は、それぞれ、モータドライバ部７０を構成するｐ個のモータドライバ９０Ｍに対して各信号を並列に供給するように接続されている。
　モータドライバ部７０の各モータドライバ９０Ｍには、演出制御ＣＰＵ２００がスレーブアドレスとして用いるデバイスＩＤ（個々のモータドライバ９０Ｍの識別子）が設定されている。説明上、仮に、図示のように各モータドライバ９０ＭのデバイスＩＤ（スレーブアドレス）をｍｔ１～ｍｔ（ｐ）と表記する。
【００７８】
　枠ドライバ部６１及び盤ドライバ部６２における各ＬＥＤドライバ９０としては、例えば２４チャネルＬＥＤドライバである「ＬＶ５２３６Ｖ（三洋半導体株式会社製）」を用いることができ、２４個の電流端子を備える。従って１つのＬＥＤドライバ９０によっては、最大２４個の系列にＬＥＤ駆動電流を供給することができる。
　なお、ここでは１つの「系列」とは、１つの電流端子に対して接続される１つのＬＥＤを指すものとする。但し、「系列」は、１つの電流端子に対して直列又は並列で接続される複数個のＬＥＤの群を指し得るものである。

枠ドライバ部６１におけるＬＥＤドライバ９０の数ｎは、枠側に配置されるＬＥＤ系列数（発光部２０ｗの系列数）によって決められる。従ってｎは１の場合もあるし、２以上の場合もある。枠ドライバ部６１は１又は複数のＬＥＤドライバ９０を有する。
また盤ドライバ部６２におけるＬＥＤドライバ９０の数ｍは、盤側に配置されるＬＥＤ系列数（発光部２０ｂの系列数）によって決められる。従ってｍは１の場合もあるし、２以上の場合もある。盤ドライバ部６２は１又は複数のＬＥＤドライバ９０を有する。
【００７９】
図５ＡにＬＥＤドライバ９０の要部の概略構成例を示す。
ＬＥＤドライバ９０は、シリアルバスインターフェース９１、アドレス設定部９２、データバッファ／ＰＷＭコントローラ９３、Ｄ／Ａ変換器９４、駆動電流源回路９５－１～９５－２４を備える。
駆動電流源回路９５－１～９５－２４は、上記の２４系列の駆動電流出力を、それぞれ電流端子９６－１～９６－２４から行う電流源である
【００８０】
このＬＥＤドライバ９０には、シリアルバスインターフェース９１に対し、演出制御ＣＰＵ２００からのイネーブル信号ENABLE、クロック信号CLK、シリアルデータDATAが入力される。シリアルバスインターフェース９１は、イネーブル信号ENABLEで規定される期間に、クロック信号CLKのタイミングでシリアルデータDATAを取り込む。シリアルバスインターフェース９１は、取り込んだシリアルデータをパラレルデータに変換してデータバッファ／ＰＷＭコントローラ９３に転送する。
なおシリアルデータの形式については図３０で後述するが、演出制御ＣＰＵ２００はスレーブアドレスを指定してＬＥＤ駆動データを送信してくる。
データバッファ／ＰＷＭコントローラ９３は、シリアルバスインターフェース９１から転送されたパラレルデータについて、スレーブアドレス確認を行う。パラレルデータに含まれるスレーブアドレスが、アドレス設定部９２に設定された自己のスレーブアドレス（ｗ１～ｗ（ｎ）、ｂ１～ｂ（ｍ）のいずれか）と一致していることを確認した場合に、該パラレルデータに含まれるＬＥＤ駆動データを有効なデータとして、指定されたレジスタに格納する。
【００８１】
データバッファ／ＰＷＭコントローラ９３は、各系列のＬＥＤ駆動データを取り込んだら、そのＬＥＤ駆動データで示された輝度情報（階調値）に応じた値を、２４系列の各駆動制御値としてＤ／Ａ変換器９４に出力する。
Ｄ／Ａ変換器９４は、輝度情報に応じた値をアナログ信号に変換し、各電流源回路９５－１～９５－２４への制御信号とする。
【００８２】
電流端子９６－１～９６－２４の全部（又は一部）には２４系列のＬＥＤ１２０が接続される。
各系列（電流端子９６－１～９６－２４）では、ＬＥＤ１２０及び抵抗Ｒの直列接続に対して電源電圧Ｖｃｃが印加される。電流源回路９５－１～９５－２４によって各系列のＬＥＤ１２０に電流が流され、発光が行われる。
即ち各電流源回路９５－１～９５－２４は、Ｄ／Ａ変換器９４から供給された信号に応じた電流量の駆動電流を、対応する系列のＬＥＤ１２０に流すように動作する。
【００８３】
このようなＬＥＤ駆動制御を、１つの系列について具体的にいうと、データバッファ／ＰＷＭコントローラ９３は、当該系列の階調値に応じたパルスデューティに相当するデジタルデータ列をＤ／Ａ変換器９４に出力し、Ｄ／Ａ変換器９４は、デジタルデータ列をアナログ信号としてのパルス信号に変換して当該系列の電流源回路９５に供給する。電流源回路９５はパルス信号のＨ／Ｌにより出力制御され、例えば０ｍＡと５ｍＡの電流出力を行う。例えばこのような動作で、結果的に階調値に応じた平均電流値となる駆動電流がＬＥＤ１２０に流れることとなる。

なお、本実施形態では、ＰＷＭ駆動方式により、電流値が例えば０ｍＡと５ｍＡとされ、時間軸方向で（積分的に）階調制御がされるものとしているが、もちろん階調制御はこれに限らず、実際に電流値を階調に応じて変化させても良いことはいうまでもない。デューティ制御であろうと、レベル制御であろうと、あくまでも単位時間あたりの平均電流値が階調に応じたレベルとされることで適切な階調表現が可能となる。

【００８４】

モータドライバ部７０における各モータドライバ９０Ｍについても、ＬＥＤドライバ９０と同様の構成でよい。例えば同じく「ＬＶ５２３６Ｖ（三洋半導体株式会社製）」を用いることができる。

モータドライバ部７０におけるモータドライバ９０Ｍの数ｐは、可動体役物のモータ数によって決められる。従ってｐは１の場合もあるし、２以上の場合もある。モータドライバ部７０は１又は複数のモータドライバ９０Ｍを有する。

【００８５】

モータドライバ９０Ｍには、可動体役物を駆動する可動体役物モータ６５が接続される。

図５Ｂに、或るモータドライバ９０Ｍの電流端子９６－１～９６－２４の全部（又は一部）に可動体役物モータ６５としての例えばステッピングモータ１２１が接続された例を示している。図５Ｂは電流端子９６－１～９６－２４の部分のみを示しているが、モータドライバ９０Ｍの内部構成は図５Ａと同様である。

ここでは４相のステッピングモータ１２１に対してそれぞれ、電流端子９６－１～９６－４、電流端子９６－５～９６－８、・・・電流端子９６－２１～９６－２４により駆動電流を供給する構成例を示している。

図５Ａに示した構成のドライバは、与えられたコマンド（シリアルデータ）によって指示される電流を電流端子９６－１～９６－２４から出力する回路であることから、図５Ｂのようにステッピングモータやソレノイド等の物理的可動体駆動デバイスに対するモータドライバ９０Ｍとしても使用することができる。

可動体役物の動作は演出シナリオによって細かく設定され、それに応じて演出制御部５１は駆動方向や駆動量などを制御するわけであるが、モータドライバ部７０におけるモータドライバ９０Ｍを利用して可動体役物を駆動することで、ランプ部６４の各発光部（ＬＥＤ２０ｂ）とともにシリアルデータによる可動体役物制御が可能となり、制御処理及び構成が効率化できる。

【００８６】

なお、１つのＬＥＤドライバ９０において、一部の電流端子がＬＥＤ駆動に用いられ、他の一部の電流端子がステッピングモータやソレノイド等の駆動に用いられるという手法を採っても良い。

【００８７】

以上のように、ＬＥＤドライバ９０は、演出制御ＣＰＵ２００から受信したシリアルデータDATAに応じて、指定された輝度で各ＬＥＤ１２０が発光されるように駆動することとなる。またモータドライバ９０Ｍは、シリアルデータDATAに応じて、ステッピングモータ１２１を駆動する。

そして第１系統、第２系統、第３系統の駆動信号出力手段（枠ドライバ部６１、盤ドライバ部６２、モータドライバ部７０）には、それぞれが１又は複数の演出手段に駆動信号を出力する複数のＬＥＤドライバ９０又はモータドライバ９０Ｍが含まれ、１つの系統内の各ＬＥＤドライバ９０やモータドライバ９０Ｍには、演出制御部５１からのシリアルデータが、並列に送信される構成とされている。そして各ＬＥＤドライバ９０やモータドライバ９０Ｍは、自己のＩＤ（スレーブアドレス）が含まれる駆動データ（シリアルデータ）を取得する。

後述するが本実施形態では、演出制御部５１（演出制御ＣＰＵ２００）は、発光制御に関しては、一送信単位のシリアルデータを、第１系統、第２系統の駆動信号出力手段（枠ドライバ部６１、盤ドライバ部６２）に対して略同時的に出力し、該送信単位のシリアル

データについての、枠ドライバ部６１、盤ドライバ部６２に対する送信完了後に、次の一送信単位のシリアルデータを、枠ドライバ部６１、盤ドライバ部６２に対して略同時的に出力する送信処理を行うようにしている。
【００８８】
［1-5. 主制御部の処理］

以下、本実施形態の制御処理につき説明する。まずここでは主制御部（主制御基板）５０によるメイン処理について述べる。
図６は、主制御部５０のメイン処理を示すフローチャートである。メイン処理が開始されるのは、停電状態からの復旧時のように初期化スイッチ（図示せず）が操作されることなく電源がＯＮ状態になる場合と、初期化スイッチがＯＮ操作されて電源がＯＮ状態になる場合とがある。いずれの場合でも、パチンコ遊技機１に電源が投入されると、電源基板５８によって各制御基板に電圧が供給される。この場合に主制御部５０（主制御ＣＰＵ１００）は図６に示すメイン処理を開始する。
【００８９】
この主制御側メイン処理において、主制御ＣＰＵ１００はステップＳ１１で、まず遊技動作開始前における必要な初期設定処理を実行する。例えば最初に自らを割込み禁止状態に設定すると共に、所定の割込みモード（割込みモード２）に設定し、またマイクロコンピュータの各部を含めてＣＰＵ内部のレジスタ値を初期設定する。
次に主制御ＣＰＵ１００はステップＳ１２で、図示してない入力ポートを介して入力されるＲＡＭクリアスイッチの出力信号であるＲＡＭクリア信号の状態（ＯＮ、ＯＦＦ）を判定する。ＲＡＭクリア信号とは、ＲＡＭの全領域を初期設定するか否かを決定する信号である。ＲＡＭクリア信号としては通常、パチンコ店の店員が操作する初期化スイッチのＯＮ／ＯＦＦ状態に対応した値を有している。
【００９０】
ＲＡＭクリア信号がＯＮ状態であった場合、主制御ＣＰＵ１００は処理をステップＳ１２からＳ１６に進め、ＲＡＭの全領域のゼロクリアを行う。したがって、電源遮断時にセットされたバックアップフラグの値は、他のチェックサム値などと共にゼロとなる。
続いてステップＳ１７で主制御ＣＰＵ１００は、ＲＡＭ領域がゼロクリアされたことを報知するための「ＲＡＭクリア表示コマンド」を初期化コマンドとして各制御基板に送信する。そしてステップＳ１８で、ＲＡＭクリア報知タイマに、ＲＡＭクリアされた旨を報知するための時間として、例えば３０秒を格納する。
【００９１】
次に主制御ＣＰＵ１００はステップＳ１９で、タイマ割込み動作を起動する割込み信号を出力するＣＴＣを初期設定して、ＣＰＵを割込み許可状態に設定する。
その後はステップＳ２０、Ｓ２１、Ｓ２２の処理として、割込みが発生するまで割込禁止状態と割込許可状態とを繰り返すとともに、その間に、各種乱数更新処理を実行する。このステップＳ２１の各種乱数更新処理では、特別図柄変動表示や普通図柄変動表示に使用される各種乱数の初期値（スタート値）変更のために使用する乱数や、変動パターンの選択に利用される変動パターン用乱数を更新する。
なお、特別図柄変動表示や普通図柄変動表示に使用される各種乱数とは、例えばインクリメント処理によって所定数値範囲を循環している大当り抽選に係る乱数（図柄抽選に利用される特別図柄判定用乱数）や、補助当り抽選に係る乱数（補助当りの当落抽選に利用される補助当り判定用乱数）などである。また初期値変更のために使用する乱数とは、特別図柄判定用初期値乱数、補助当り判定用初期値乱数などである。
【００９２】
主制御ＲＡＭ１０２には大当り抽選に係る図柄抽選、補助当り抽選、または変動パターン抽選などに利用される各種の乱数カウンタとして、特別図柄判定用乱数カウンタ初期値の生成用カウンタ、特別図柄判定用乱数カウンタ、補助当り判定用乱数カウンタ初期値の生成用カウンタ、補助当り判定用乱数カウンタ、変動パターン用乱数１カウンタ、変動パ

ターン用乱数２カウンタなどが設けられている。これらのカウンタは、ソフトウェア的に乱数を生成する乱数生成手段としての役割を果たす。

ステップＳ２１の各種乱数更新処理では、上述の特別図柄判定用乱数カウンタや補助当り判定用乱数カウンタの初期値を生成する２つの初期値生成用カウンタ、変動パターン用乱数１カウンタ、変動パターン用乱数２カウンタなどを更新して、上記各種のソフト乱数を生成する。例えば変動パターン用乱数１カウンタとして取り得る数値範囲が０～２３８とすると、主制御ＲＡＭ１０２の変動パターン用乱数１の値を生成するためのカウント値記憶領域から値を取得し、取得した値に１を加算してから元のカウント値記憶領域に格納する。このとき、取得した値に１を加算した結果が２３９であれば０を元の乱数カウンタ記憶領域に格納する。他の初期値生成用乱数カウンタも同様に更新する。主制御ＣＰＵ１００は、間欠的に実行されるタイマ割込処理を行っている間を除いて、各種乱数更新処理を繰り返し実行するようになっている。

【００９３】

以上はステップＳ１２でＲＡＭクリアスイッチＯＮと判定された場合について述べた。ＲＡＭクリアスイッチＯＦＦの場合を続いて説明する。例えば停電状態からの復旧時には、初期化スイッチ（ＲＡＭクリア信号）はＯＦＦ状態である。このような場合、主制御ＣＰＵ１００はステップＳ１２からＳ１３に処理を進め、バックアップフラグ値を判定する。なお、バックアップフラグは、電源遮断時にＯＮ状態に設定され、電源復帰後の最初のタイマ割込み処理でＯＦＦ状態にリセットされるよう構成されている。

したがって、電源投入時や停電状態からの復旧時である場合には、通常では、バックアップフラグがＯＮ状態のはずである。ただし、何らかの理由で電源遮断までに所定の処理が完了しなかったような場合には、バックアップフラグはリセット（ＯＦＦ）状態になる。そこで、バックアップフラグがＯＦＦ状態である場合には、主制御ＣＰＵ１００は処理をステップＳ１３からＳ１６に進め、遊技機の動作を初期状態に戻す。

【００９４】

一方、バックアップフラグがＯＮ状態であれば、主制御ＣＰＵ１００は処理をステップＳ１３からＳ１４に進め、チェックサム値を算出するためのチェックサム演算を実行する。ここで、チェックサム演算とは、主制御ＲＡＭ１０２のワーク領域を対象とする８ビット加算演算である。

そして、チェックサム値が算出されたら、この演算結果を、主制御ＲＡＭ１０２のＳＵＭ番地の記憶値と比較をする。このＳＵＭ番地には、電源遮断時に、同じチェックサム演算によるチェックサム値が記憶されている。そして、記憶された演算結果は、主制御ＲＡＭ１０２の他のデータと共に、バックアップ電源によって維持されている。したがって、本来は、ステップＳ１４の判定によって両者が一致するはずである。

しかし、電源遮断時にチェックサム演算が実行できなかった場合や、実行できても、その後、メイン処理のチェックサム演算の実行時までの間に、ワーク領域のデータが破損している場合もある。このような場合にはステップＳ１４の判定結果は不一致となる。

判定結果の不一致によりデータ破損が検出された場合には、主制御ＣＰＵ１００はステップＳ１４からＳ１６の処理に進んでＲＡＭクリア処理を実行し、遊技機の動作状態を初期状態に戻す。

【００９５】

ステップＳ１４でのチェックサム演算によるチェックサム値と、ＳＵＭ番地の記憶値とが一致する場合には、主制御ＣＰＵ１００はステップＳ１５に進み、バックアップデータに基づき、電源遮断前におけるスタックポインタを復帰し、電源遮断時の処理状態から遊技を開始するために必要な遊技復旧処理を実行する。

そしてステップＳ１５の遊技復旧処理を終えると、ステップＳ１９の処理に進み、ＣＴＣを初期設定してＣＰＵを割込み許可状態に設定し、その後は、割込みが発生するまで割込禁止状態と割込許可状態とを繰り返すとともに、その間に、上述した各種乱数更新処理を実行する（ステップＳ２０～Ｓ２２）。

【００９６】

次に主制御ＣＰＵ１００のタイマ割込処理について説明する。図７に主制御ＣＰＵ１００のタイマ割込処理を示している。この主制御タイマ割込処理は、ＣＴＣからの一定時間（４ｍｓ程度）ごとの割込みで起動され、上述したメイン処理実行中に割り込んで実行される。

【００９７】

タイマ割込みが生じると、主制御ＣＰＵ１００はレジスタの内容をスタック領域に退避させた後、まず図７のステップＳ５１として電源基板５８からの電源の供給状態を監視する電源異常チェック処理を行う。この電源異常チェック処理では、主に、電源が正常に供給されているかを監視する。ここでは、例えば電断が生じるなどの異常が発生した場合、電源復帰時に支障なく遊技を復帰できるように、電断時における所定の遊技情報をＲＡＭに格納するバックアップ処理などが行われる。

【００９８】

次にステップＳ５２で、主制御ＣＰＵ１００は遊技動作制御に用いられるタイマを管理するタイマ管理処理を行う。パチンコ遊技機１の遊技動作制御に用いる各種タイマ（たとえば特別図柄役物動作タイマなど）のタイマ値は、この処理で管理（更新）される。

【００９９】

ステップＳ５３では、主制御ＣＰＵ１００は入力管理処理を行う。この入力管理処理では、パチンコ遊技機１に設けられた各種センサによる検出情報を入賞カウンタに格納する。ここでの各種センサによる検出情報とは、例えば上始動口センサ７１、下始動口センサ７２、ゲートセンサ（普通図柄始動口センサ）７３、第１大入賞口センサ７５、第２大入賞口センサ７６、一般入賞口センサ７４などの入賞検出スイッチから出力されるスイッチ信号のＯＮ／ＯＦＦ情報（入賞検出情報）である。

このステップＳ５３の処理により、各入賞口において入賞を検出（入賞が発生）したか否かが割込みごとに監視される。また上記「入賞カウンタ」とは、各々の入賞口ごとに対応して設けられ、入賞した遊技球数（入賞球数）を計数するカウンタである。本実施形態では、主制御ＲＡＭ１０２の所定領域に、上始動口４１用の上始動口入賞カウンタ、下始動口４２ａ用の下始動口入賞カウンタ、ゲート４４用の普通図柄始動口入賞カウンタ、第１大入賞口４５ａ用の第１大入賞口入賞カウンタ、第２大入賞口４６ａ用の第２大入賞口入賞カウンタ、一般入賞口４３用の一般入賞口用の入賞カウンタなどが設けられている。

またこの入力管理処理では、入賞検出スイッチからの検出情報が入賞を許容すべき期間中に入賞したか否かに基づいて、不正入賞があったか否かも監視される。たとえば大当り遊技中でないにもかかわらず第１、第２大入賞口センサ７５，７６が遊技球を検出したような場合は、これを不正入賞とみなして入賞検出情報を無効化し、その無効化した旨を外部に報知するべく後述のステップＳ５５のエラー管理処理において所定のエラー処理が行われるようになっている。

【０１００】

ステップＳ５４では、主制御ＣＰＵ１００は各変動表示に係る乱数を定期的に更新するタイマ割込内乱数管理処理を行う。この定期乱数更新処理では、特別図柄判定用乱数や補助当り判定用乱数の更新（割込み毎に＋１加算）と、乱数カウンタが一周するごとに、乱数カウンタのスタート値を変更する処理を行う。例えば特別図柄判定用乱数カウンタの値を所定範囲で更新（＋１加算）し、特別図柄判定用乱数カウンタが１周するごとに、特別図柄判定用乱数カウンタ初期値の生成用カウンタの値を読み出し、その生成用カウンタの値を特別図柄判定用乱数カウンタに格納する。これにより、特別図柄判定用乱数カウンタのスタート値が上記の生成用カウンタの値に応じて変更されるので、更新周期は一定でありながらも特別図柄判定用乱数カウンタのカウント値はランダムになる。

【０１０１】

ステップＳ５５では、主制御ＣＰＵ１００は、遊技動作状態の異常の有無を監視するエラー管理処理を行う。このエラー管理処理では、遊技動作状態の異常として、例えば基板間に断線が生じたか否かの監視や、不正入賞があったか否かの監視などをして、これらの動作異常（エラー）が発生した場合には、そのエラーに対応した所定のエラー処理を行う

。
エラー処理としては、例えば所定の遊技動作（例えば遊技球の払い出し動作や遊技球の発射動作など）の進行を停止させたり、エラー報知用コマンドを演出制御部５１に送信して、演出手段によりエラーが発生した旨を報知させたりする。
【０１０２】
ステップＳ５６では、主制御ＣＰＵ１００は賞球管理処理を行う。この賞球管理処理では、ステップＳ５３の入力管理処理で格納したデータを把握して、上述の入賞カウンタの確認を行い、入賞があった場合は、賞球数を指定する払出制御コマンドを払出制御基板５３に送信する。
この払出制御コマンドを受信した払出制御基板５３は、遊技球払出装置５５を制御し、指定された賞球数の払い出し動作を行わせる。これにより、それぞれの入賞口に対応した賞球数が払い出されるようになっている。入賞口に対応した賞球数とは、入賞口別に設定された入賞球１個当りの所定の賞球数×入賞カウンタの値分の賞球数である。
【０１０３】
ステップＳ５７では主制御ＣＰＵ１００は、普通図柄管理処理を行う。この普通図柄管理処理では、普通図柄変動表示における補助当り抽選を行い、その抽選結果に基づいて、普通図柄の変動パターンや普通図柄の停止表示態様を決定したり、所定時間毎に点滅を繰り返す普通図柄のデータ（普通図柄変動中のＬＥＤ点滅表示用データ）を作成したり、普通図柄が変動中でなければ、停止表示用のデータ（普通図柄停止表示中のＬＥＤ点滅表示用データ）を作成したりする。
【０１０４】
ステップＳ５８では、主制御ＣＰＵ１００は、普通電動役物管理処理を行う。この普通電動役物管理処理では、ステップＳ５７の普通図柄管理処理の補助当り抽選の抽選結果に基づき、普通電動役物ソレノイド７７に対するソレノイド制御用の励磁信号の生成およびそのデータ（ソレノイド制御データ）の設定を行う。ここで設定されたデータに基づき、後述のステップＳ６４のソレノイド管理処理にて、励磁信号が普通電動役物ソレノイド７７に対して出力され、これにより可動翼片４２ｂの動作が制御される。
ステップＳ５９では、主制御ＣＰＵ１００は、特別図柄管理処理を行う。この特別図柄管理処理では、主に、特別図柄変動表示における大当り抽選を行い、その抽選結果に基づいて、特別図柄の変動パターン（先読み変動パターン、変動開始時の変動パターン）や特別停止図柄などを決定する。
ステップＳ６０では、主制御ＣＰＵ１００は特別電動役物管理処理を行う。この特別電動役物管理処理では、主に、大当り抽選結果が「大当り」または「小当り」であった場合、その当りに対応した当り遊技を実行制御するために必要な設定処理を行う。
【０１０５】
ステップＳ６１では、主制御ＣＰＵ１００は右打ち報知情報管理処理を行う。この右打ち報知情報管理処理では、例えば第１、第２大入賞口４５ａ，４６ａが開放される機会や可動翼片４２ｂが駆動される電サポ状態など、右打ちが有利な状況において右打ち指示報知を行う「発射位置誘導演出（右打ち報知演出）」を現出させるための処理を行う。右打ち指示とは、具体的には、右遊技領域３ｃを狙う旨を有技者に指示する演出動作であり、例えば液晶表示装置３２に「右打ち」を遊技者に促す画像を表示させたり、スピーカ２５から右打ちメッセージ音声を発生させる。
右打ち報知演出が行われる場合、この右打ち報知情報管理処理において、演出制御コマンドとして、右打ち報知演出の実行指示する「右打ち指示コマンド」が演出制御部５１に送信され、このコマンドを受けて、演出制御部５１が、画像や音声による右打ち報知の実行制御を行う。
ステップＳ６２では、主制御ＣＰＵ１００は、ＬＥＤ管理処理を行う。このＬＥＤ管理処理は、図柄表示部３３に対して普通図柄表示や第１，第２特別図柄表示のための表示データを出力する処理である。この処理により、普通図柄や特別図柄の変動表示および停止表示が行われる。なお、ステップＳ５７の普通図柄管理処理で作成された普通図柄の表示

データや、ステップＳ５９の特別図柄管理処理中の特別図柄表示データ更新処理で作成される特別図柄の表示データは、このＬＥＤ管理処理で出力される。

【０１０６】

ステップＳ６３では、主制御ＣＰＵ１００は、外部端子管理処理を行う。この外部端子管理処理では、枠用外部端子基板５７を通して、パチンコ遊技機１の動作状態情報をホールコンピュータや島ランプなどの外部装置に対して出力する。動作状態情報としては、大当り遊技が発生した旨（条件装置が作動した旨）、小当り遊技が発生した旨、図柄変動表示が実行された旨（特別図柄変動表示ゲームの開始または終了した旨）、入賞情報（始動口や大入賞口に入賞した旨や賞球数情報）などの情報が含まれる。

ステップＳ６４では、主制御ＣＰＵ１００は、ソレノイド管理処理を行う。このソレノイド管理処理では、ステップＳ５８の普通電動役物管理処理で作成されたソレノイド制御データに基づく普通電動役物ソレノイド７７に対する励磁信号の出力処理や、ステップＳ６０の特別電動役物管理処理で作成されたソレノイド制御データに基づく第１，第２大入賞口ソレノイド７８，７９に対する励磁信号の出力処理を行う。これにより、可動翼片４２ｂや開放扉４５ｂ、４６ｂが所定のパターンで動作し、下始動口４２ａや大入賞口４５ａ、４６ｂが開閉される。

【０１０７】

主制御ＣＰＵ１００は、以上のステップＳ５１～ステップＳ６４の処理を終えた後、退避していたレジスタの内容を復帰させて、ステップＳ６５で割込み許可状態に設定する。これにより、タイマ割込処理を終了して、割込み前の図６の主制御側メイン処理に戻り、次のタイマ割込みが発生するまで主制御メイン処理を行う。

【０１０８】

次に、図７のステップＳ５９の特別図柄管理処理について説明する。図８は、ステップＳ５９の特別図柄管理処理の詳細を示すフローチャートである。

図８において、主制御ＣＰＵ１００はステップＳ７１で、まず第１特別図柄側（上始動口４１側）に関する第１特図始動口チェック処理を行い、次いでステップＳ７２で第２特別図柄側（下始動口４２ａ側）に関する第２特図始動口チェック処理を行う。

【０１０９】

次に主制御ＣＰＵ１００はステップＳ７３で、小当り中フラグの状態を判定する。この小当り中フラグとは、小当り遊技中であるか否かを指定するためのフラグで、当該フラグがＯＮ状態である場合には小当り遊技中である旨を示し、当該フラグがＯＦＦ状態である場合には小当り遊技中ではない旨を示す。

【０１１０】

小当り中フラグがＯＦＦ状態の場合、主制御ＣＰＵ１００はステップＳ７４で、条件装置作動フラグの状態を判定する。この条件装置作動フラグとは、大当り遊技中であるか否かを指定するフラグで、当該フラグがＯＮ状態である場合には大当り遊技中である旨を示し、当該フラグがＯＦＦ状態である場合には大当り遊技中ではない旨を示す。

【０１１１】

条件装置作動フラグがＯＦＦ状態の場合、主制御ＣＰＵ１００はステップＳ７５で、特別図柄の挙動を示す処理状態に応じた処理を行うために処理を分岐する。

すなわち、小当り遊技中でもなく大当り遊技中でもないとしてステップＳ７５に進んだ場合、特別図柄が変動可能状態下に置かれ、これにより、特別図柄動作ステータスが「待機中」「変動中」「確認中」のいずれの状態であるかに応じて、それぞれに対応する特別図柄の変動表示動作に関する処理が行われる。

一方、小当り遊技中、または大当り遊技中である場合には、特別図柄が変動可能状態下には置かれないため、特別図柄の変動表示動作に関する処理を行わずにステップＳ７９に進む。

ここで、上述の「待機中」とは、特別図柄の挙動が次回変動のための待機状態である旨を示し、「変動中」とは特別図柄の挙動が変動（変動表示）中である旨を示し、「確認中」とは特別図柄の変動が終了して停止（確定）表示中（特別図柄確認時間中）である旨を

示す。
【０１１２】
具体的には、特別図柄動作ステータスが「待機中」である場合には、ステップＳ７５からＳ７６に進み、特別図柄変動開始処理を、「変動中」である場合には、ステップＳ７５からＳ７７に進み特別図柄変動中処理を、「確認中」である場合には、ステップＳ７５からＳ７８に進み特別図柄確認時間中処理を行う。これらの処理により、特別図柄変動表示ゲームとして、特別図柄の変動開始および変動停止を一セットする変動表示動作が実現される。
【０１１３】
主制御ＣＰＵ１００はステップＳ７９では、特別図柄表示データ更新処理を行う。この特別図柄表示データ更新処理では、特別図柄が変動中であるか否かを判定し、変動中であれば、所定時間（たとえば０．５秒）毎に点滅を繰り返す特別図柄のデータを作成し、特別図柄が変動中でなければ、停止表示用のデータを作成する。ここで作成した特別図柄の表示データは、図７のＬＥＤ管理処理（ステップＳ６２）で出力される。
【０１１４】
続いて、上記したステップＳ７１の第１特図始動口チェック処理について説明する。
図９は、ステップＳ７１の第１特図始動口チェック処理の詳細を示すフローチャートである。第１特図始動口チェック処理は、所定の始動条件の成立に基づいて実行される入賞時処理としての役割を果たし、第１特別図柄変動表示ゲームに係る上始動口４１についての入賞チェック処理が行われる。
【０１１５】
主制御ＣＰＵ１００は、まずステップＳ７１で上始動口４１において入賞（入賞球）を検出したか否かを判定する。上始動口４１の入賞を検出した場合、ステップＳ８２で第１特別図柄の保留球数が４以上となる場合であるか否か判定する。すなわち、第１特別図柄の保留球数（以下「第１特図保留球」と称する）が最大保留記憶数（ここでは上限４個）未満であるか否かを判定する。
なお、ステップＳ８１で上始動口４１の入賞を検出しなかった場合、何もしないでこの第１特図始動口チェック処理を抜ける。
【０１１６】
第１特図保留球数が４以上である場合には、ステップＳ９５に進む。一方、保留球数が４未満の場合はステップＳ８３で保留球数を１加算（＋１）する。
次にステップＳ８４で、第１特図保留球に基づく、第１特別図柄変動表示ゲームに利用される各種乱数（大当り判定用乱数、特別図柄判定用乱数、変動パターン用乱数１、および変動パターン用乱数２）の乱数カウンタの現在値を取得して、それぞれ主制御ＲＡＭ１０２の保留記憶エリアに格納する。この保留記憶エリアには、取得された上述の各種乱数値（図柄遊技に係る所定の遊技情報）が保留データとして、第１特別図柄の変動表示動作に供されるまで所定の記憶数を上限として始動条件の成立順（入賞順。ただし、本実施形態では、最大保留記憶数である第１特図保留球数が４未満の場合）に保留記憶される。
なお主制御ＲＡＭ１０２の保留記憶エリアには、第１特別図柄と第２特別図柄とに対応した保留記憶エリアが設けられており、それぞれ保留球数分の保留データを格納可能となっている。
【０１１７】
特別図柄判定用乱数、変動パターン用乱数１、および変動パターン用乱数２は、それぞれに対応した乱数カウンタから抽出される。これら乱数値は、主制御ＲＡＭ１０２に設けられた各々に対応したカウント値記憶領域において、間欠的に生じる割込み処理と割込み処理の間に実行されるメイン処理において乱数的に更新されているので、その値がそのまま取得される。他方、大当り判定用乱数値は、ハードウェア的に乱数を生成する乱数生成回路（大当り判定用乱数カウンタ：図示せず）から抽出される。この乱数生成回路は、大当り判定用乱数値をカウントしているカウンタである。乱数生成回路は、上始動口センサ７１または下始動口センサ７２から遊技球の検出信号が入力されると、このタイミングに

おけるカウント値をラッチし、ラッチした値を乱数生成回路に内蔵されたレジスタに記憶し、当該カウント値がマイクロコンピュータに入力されるようになっている。主制御ＣＰＵ１００は、このカウント値を大当り判定用乱数値として、主制御ＲＡＭ１０２の保留記憶エリアと別個に設けられた所定の記憶領域（特別電動役物作動判定用乱数ラッチバッファ）に格納し、必要なタイミングで特別電動役物作動判定用乱数ラッチバッファから格納された大当り判定用乱数値を読み出し利用する。

本実施形態では、入賞時における大当り判定用乱数値を特別電動役物作動判定用乱数ラッチバッファと該当する保留記憶エリアとに格納するようになっているが、特別電動役物作動判定用乱数ラッチバッファを設けない場合は、保留記憶エリアにそのまま格納し、必要なタイミングで、該当保留記憶エリアの大当り判定用乱数値を読み出すように構成しても良い。

【０１１８】

次いでステップＳ８６で、保留加算コマンドを作成するための入賞コマンドデータとして、先読み判定を禁止する先読み禁止データを取得し、第１特別図柄側の先読み判定を禁止する「第１特図先読み禁止条件」が成立しているか否かを判定する。なお、第１特図先読み禁止条件とは、第１特別図柄に関する先読み予告、つまり第１特図保留球を対象とした先読み予告を禁止する条件である。本実施形態では、遊技状態が時短状態か、または確変状態Ａ～Ｃのいずれかであることを禁止条件とし、これらいずれかの遊技状態である場合、第１特別図柄に関する先読み予告を禁止（先読み予告抽選自体を禁止）する。

【０１１９】

第１特図先読み禁止条件が成立している場合にはステップＳ９５に進む。

第１特図先読み禁止条件が成立していない場合にはステップＳ８７に進み、第１特図用当り乱数判定テーブルを取得し、ステップＳ８８で主制御ＲＡＭ１０２（特別電動役物作動判定用乱数ラッチバッファ）に格納された大当り判定用乱数値を取得して、ステップＳ８９の乱数判定処理を行う。

乱数判定処理では、大当り判定用乱数値を利用し、特別図柄変動表示ゲーム（ここでは、第１特別図柄変動表示ゲーム）に係る「当落抽選」を行う。

【０１２０】

次にステップＳ９０で主制御ＣＰＵ１００は、第１特図用図柄テーブル選択テーブルを取得し、ステップＳ９１で主制御ＲＡＭ１０２に格納された特別図柄判定用乱数値を取得して、ステップＳ９２で特別停止図柄データ作成処理を行う。

特別停止図柄データ作成処理では、乱数判定処理の当落抽選結果と特別図柄判定用乱数値を利用し、特別図柄変動表示ゲーム（ここでは、第１特別図柄変動表示ゲーム）に係る「図柄抽選」を行う。

【０１２１】

ステップＳ９３では主制御ＣＰＵ１００は始動口入賞時乱数判定処理を行う。この始動口入賞時乱数判定処理では、ステップＳ９２の停止図柄データ作成処理で得られたデータ（特別図柄判定データと変動パターン選択用オフセット）に基づき、保留加算コマンドの作成に利用される入賞コマンドデータ（２バイト目（ＥＶＥＮＴ）：下位バイト）を取得

そしてステップＳ９４で入賞コマンドデータ（下位バイト（ＥＶＥＮＴ））を作成し、またステップＳ９５で保留球数（上位バイト（ＭＯＤＥ））に基づく保留加算コマンドを作成し、ステップＳ９６で演出制御部５１に送信する。

これにより、第１特図始動口チェック処理を抜けて、図８のステップＳ７２の第２特別図柄始動口チェック処理を行う。第２特別図柄始動口チェック処理は、第２特別図柄に関して、図９と概略同様の処理を行うものであるため詳細な説明は省略する。

【０１２２】

ここまで主制御部５０（主制御ＣＰＵ１００）のメイン処理、タイマ割込処理、及び特別図柄管理処理について説明したが、主制御部５０は逐次必要なコマンドを演出制御部５１に対して送信する。

コマンドの例として、特別図柄変動表示ゲームの当落抽選にかかる変動パターンコマン

ド、入賞時コマンドの一部を図１０Ａに示し、また図柄指定コマンドの一部を図１０Ｂに示している。

変動パターンコマンド、入賞時コマンドは、上位バイトと下位バイトで変動パターンの種別が規定され、外れ／当たりに応じて上位バイトが異なるように構成されている。

図柄指定コマンドは、上位バイトで第１、第２特別図柄が示され、下位バイトで当たりの種別が指定される構成となっている。

【０１２３】

［1-6．演出制御部の処理］

（1-6-1：メイン処理）

続いて演出制御部５１の処理について説明する。演出制御部５１の処理としては、主に、メインループ上で１６ｍｓ毎に行われる処理（以下「１６ｍｓ処理」ともいう）と、１ｍｓ毎に行われる割り込み処理（以下「１ｍｓタイマ割込処理」ともいう）がある。

まずここでは１６ｍｓ処理を含むメイン処理について説明する。

図１１は、演出制御部５１のメイン処理を示している。演出制御部５１（演出制御ＣＰＵ２００）は、遊技機本体に対して電源が投入されると、図１１のメイン処理を開始する。

このメイン処理において、演出制御ＣＰＵ２００は、まずステップＳ１０１で、遊技動作開始前における必要な初期設定処理を行う。例えば初期設定処理として、コマンド受信割込み設定、可動体役物の起点復帰処理、ＣＴＣの初期設定、タイマ割込みの許可、マイクロコンピュータの各部を含めてＣＰＵ内部のレジスタ値の初期設定などを行う。

【０１２４】

ステップＳ１０１の初期設定処理を終えると、正常動作時の処理としてステップＳ１０２～Ｓ１１９の処理を繰り返し行う。

即ちこの例では、演出制御ＣＰＵ２００はステップＳ１０２でのＩＤチェックとステップＳ１１９での乱数更新を毎ループ行うと共に、１６ｍｓ毎に、ステップＳ１０５～Ｓ１１８の処理（１６ｍｓ処理）を行う。

【０１２５】

ステップＳ１０２のＩＤチェックでは、演出制御ＣＰＵ２００はシステム上で設定されている自己或いは接続各部のＩＤの確認を行う。もし何らかの原因により、ＩＤ異常が検出された場合は、ステップＳ１０３としてシステム停止処理を行う。

ＩＤに問題のない通常時は、演出制御ＣＰＵ２００はステップＳ１０４以下の処理を行うことになる。即ち演出制御ＣＰＵ２００は、１６ｍｓ処理の実行判断のための割込みカウンタの値が「１５」より大きい値となっているか否かを判断する。この割込みカウンタは、後述する図１２の１ｍｓタイマ割込処理のステップＳ２０７でインクリメントされていくカウンタである。従って割込みカウンタの値が「１５」より大きい場合とは、１６ｍｓ処理のタイミングになっていることを意味する。

【０１２６】

演出制御ＣＰＵ２００は、割込みカウンタの値が「１５」以下であるときは、ステップＳ１０４からＳ１１９に進み、演出用ソフト乱数の更新処理を行って１回のメイン処理を終え、再びステップＳ１０２からの処理を行う。

一方、割込みカウンタの値が「１６」以上である場合は、演出制御ＣＰＵ２００はステップＳ１０５～Ｓ１１８の処理を実行し、その後、ステップＳ１１９で演出用ソフト乱数の更新処理を行って１回のメイン処理を終え、再びステップＳ１０２からの処理を行うことになる。

【０１２７】

このように割込みカウンタでカウントされる１６ｍｓ毎に、演出制御ＣＰＵ２００はステップＳ１０５からの１６ｍｓ処理を行う。

その場合、まずステップＳ１０５では、割込みカウンタをゼロリセットする。以後、再び次の１６ｍｓ処理までのカウントを行うためである。

次にステップＳ１０６で演出制御ＣＰＵ２００は、エラー処理を行う。このエラー処理としては、ＲＡＭクリアエラー中、役物エラー中、右打ちエラー中などにおけるエラー処理タイマの処理、各種エラーが発生した際のエラー報知のためのシナリオ登録処理、エラー報知後のエラーシナリオのクリア処理などを行うこととなる。

次にステップＳ１０７では、演出制御ＣＰＵ２００はデモ処理を行う。このデモ処理では、再生音の制御、デモムービー実行や役物原点補正のシナリオ登録や、そのコマンドセットなどの処理を行う。客待ち状態などでは、このデモ処理で設定されたシナリオが実行されることでデモムービー表示が実行される。

【０１２８】

ステップＳ１０８では、演出制御ＣＰＵ２００はコマンド解析処理を行う。このコマンド解析処理では、演出制御ＣＰＵ２００が、主制御部５０から供給される演出制御コマンドがコマンド受信バッファに格納されているか否かを監視し、演出制御コマンドが格納されていればこのコマンドを読み出す。そして読み出した演出制御コマンドに対応した演出制御処理を行う。詳しくは図１３～図１５で後述する。

ステップＳ１０９では、演出制御ＣＰＵ２００は入力検知処理を行う。この入力検知処理では、操作部６０の操作子（演出ボタン１１、１２、十字キー１３）の操作による入力の検知を行い、入力を検知した場合、その操作に応じた処理を行う。

ステップＳ１１０では、演出制御ＣＰＵ２００はシナリオ更新処理を行う。この処理ではメインシナリオの更新、サブシナリオの更新が行われる。その際にはランプ部６４，６５の点灯パターン登録、再生する音の登録、可動体役物の駆動のためのモータ動作の登録なども行われる。詳しくは図１８，図１９を用いて後述する。

ステップＳ１１１では、演出制御ＣＰＵ２００は音再生処理を行う。演出制御ＣＰＵ２００は、シナリオデータに基づいてワークに音チャネルとして登録されている音データに基づいて、フレーズ番号やボリューム等のデータを音源ＩＣに出力する。これによってスピーカ２５からの効果音、音楽・音声等の再生出力が行われる。

ステップＳ１１２では、演出制御ＣＰＵ２００は役物エラー処理を行う。ここでは可動体役物の原点復帰がなされていないなどの位置エラー判定などを行う。

【０１２９】

ステップＳ１１３では演出制御ＣＰＵ２００は、ＬＥＤドライバ９０の初期化を行う。ＬＥＤドライバ９０の初期化とは、ＬＥＤドライバ９０において使用しないレジスタにデフォルト値を出力する処理である。例えば枠ドライバ部６１の各ＬＥＤドライバ９０と、盤ドライバ部６２の各ＬＥＤドライバ９０に対して初期化を行う。

ステップＳ１１３では演出制御ＣＰＵ２００は、ＬＥＤ駆動データ更新を行う。ここでは、シナリオデータに基づいてワークにランプチャネルとして登録されているランプデータに基づいて、ＬＥＤ駆動データを作成する処理が行われる。詳しくは図２４を用いて後述する。

ステップＳ１１４では演出制御ＣＰＵ２００は、ＬＥＤ出力処理を行う。即ち演出制御ＣＰＵ２００は、枠ドライバ部６１の各ＬＥＤドライバ９０と、盤ドライバ部６２の各ＬＥＤドライバ９０に対して、シリアルデータ出力チャネルｃｈ１、ｃｈ２から、ＬＥＤドライバ９０に対する発光駆動データ（ＬＥＤ駆動データ）の出力を行う。詳しくは図２５を用いて後述する。

【０１３０】

ステップＳ１１６では、演出制御ＣＰＵ２００は、演出制御ＲＡＭ２０２のワーク領域を対象としてチェックサム算出及びその保存を行い、またステップＳ１１７では、バックアップデータの保存を行う。

さらにステップＳ１１８ではシナリオ更新カウンタをゼロリセットする。シナリオ更新カウンタは後述の１ｍｓタイマ割込処理でインクリメントされるカウンタである。

以上のような１６ｍｓ処理が、図１１のメインループ処理において１６ｍｓ経過毎に行われる。

【０１３１】

（1-6-2：１ｍｓタイマ割込処理）

次に図１２により１ｍｓタイマ割込処理を説明する。演出制御ＣＰＵ２００は、タイムカウントにより１ｍｓ毎に発生する割込要求に応じて、図１２の１ｍｓタイマ割込処理を実行する。

この１ｍｓタイマ割込処理においては、まずステップＳ２０１では主制御ＣＰＵ１００からのテストコマンドに応じたチェックサム算出中であるか否かを判断する。チェックサム算出中でなければ、演出制御ＣＰＵ２００はステップＳ２０２の入力処理に進む。

【０１３２】

ステップＳ２０２の入力処理とは、上述の図１１のステップＳ１０９の入力検知処理とともに操作子の操作による入力検知を行うための１ｍｓ毎の処理である。例えばこの入力処理では、操作子の操作検出信号について、信号波形にエッジが検出されたら入力カウンタをリセットし、その後、エッジが発生しない期間、入力カウンタをカウントアップしていく処理を行う。１ｍｓタイマ割込処理において、入力情報（入力信号波形のＨまたはＬ）が検知され、またエッジ有無により、入力カウンタのリセット又はインクリメントが行われる。そしてメインループ処理（１６ｍｓ処理）におけるステップＳ１０９で、入力カウンタの値が１６以上となっており、前回とは入力情報が変化している場合に、入力変化を認識するようにしている。

このような入力処理（Ｓ２０２）及び入力検知処理（Ｓ１０９）により、ノイズ・チャタリングによる入力誤認識の防止がはかられる。また、入力カウンタを用いており、本実施形態では例えば１６ビットカウンタを用いて６５５３５ｍｓ（約６５秒）までなどをカウントできるようにしているため、いわゆる長押しの検出も可能となる。

【０１３３】

ステップＳ２０３では、演出制御ＣＰＵ２００はモータ動作更新処理を行う。この場合、演出制御ＣＰＵ２００は、シナリオデータに基づいてワーク（図２０Ｃで説明するモータデータ登録情報）にモータチャネルとして登録されているモータデータに基づいて、モータ駆動データを作成する処理を行う。これは可動体役物モータ６５を駆動制御するためにモータドライバ部７０のモータドライバ９０Ｍに出力するモータ駆動データである。なお、本実施形態では、ＬＥＤ駆動データ更新は上記ステップＳ１１４により１６ｍｓ毎に行われる一方、モータ駆動データ更新は１ｍｓ毎に行われることになる。

ステップＳ２０４では、演出制御ＣＰＵ２００は、モータ駆動データを出力する。このステップＳ２０４では、可動体役物モータ６５としてのステッピングモータ１２１に対する駆動データとしてのシリアルデータを、モータドライバ部７０のモータドライバ９０Ｍに対して出力することになる。

【０１３４】

ステップＳ２０５では、演出制御ＣＰＵ２００は、割込みカウンタの値に応じて各処理を実行する。割込みカウンタは上述の１６ｍｓ処理のステップＳ１０５でゼロリセットされ、１ｍｓタイマ割込処理のステップＳ２０７でインクリメントされる。従って、１ｍｓタイマ割込処理でステップＳ２０５が実行される際には、割込みカウンタの値は０～１５のいずれかとなっている。

この割込みカウンタの値０～１５に応じて、ケース０～ケース１５としてステップＳ２０５の処理が規定される。例えば本実施形態では、ケース０ではＷＤＴ（watchdog timer）クリア信号ＯＮを行う。ケース１～３ではモータドライバ９０Ｍの初期化処理を行う。ケース８ではＷＤＴクリア信号ＯＦＦを行う。

【０１３５】

ステップＳ２０６では、演出制御ＣＰＵ２００は、液晶制御基板５２に対するコマンド（液晶制御コマンド）の出力を行う。例えば１回の１ｍｓタイマ割込処理において、１コマンド（２バイト）の送信を行う。つまりモード及びイベントとしての２バイトの演出制御コマンドが送信される。

その後、演出制御ＣＰＵ２００はステップＳ２０７で割込みカウンタのインクリメント

を行い、またステップＳ２０８でシナリオ更新カウンタのインクリメントを行う。そしてシナリオ更新カウンタの値が１００未満であれば、１ｍｓタイマ割込処理を終える。
なお、シナリオ更新カウンタは上述のように１６ｍｓ処理のステップＳ１１８でゼロリセットされるため、通常はシナリオ更新カウンタの値が１００以上となることはない。１００以上となるのは、演算異常、処理応答異常などにより１６ｍｓ処理が実行されない場合や、１６ｍｓ処理内の或る処理の進行が停止しているような場合である。このような場合は、無限ループに入り、ＷＤＴによってタイムアップ処理が行われるのを待つことになる。

【０１３６】
１ｍｓタイマ割込処理に入った際に、ステップＳ２０１でチェックサム算出中と判断された場合は、演出制御ＣＰＵ２００は処理をステップＳ２１０に進め、ＷＤＴをクリアする。そしてステップＳ２１１で液晶制御基板５２に対する演出制御コマンドの送信出力を行う。そしてステップＳ２１２で割込みカウンタをインクリメントして１ｍｓタイマ割込処理を終える。

【０１３７】
（1-6-3：コマンド解析処理）

続いて、１６ｍｓ処理として図６のステップＳ１０８で行われるコマンド解析処理について、図１３～図１５で説明する。演出制御ＣＰＵ２００は、ステップＳ１０８のコマンド解析処理として、主制御部５０から供給される演出制御コマンドがコマンド受信バッファに格納されているか否かを監視し、演出制御コマンドが格納されていればこのコマンドを読み出す。この具体的な処理として、図１３のステップＳ３０１～Ｆ３０８の処理を行う。

【０１３８】
演出制御部５１には、主制御部５０から送信されてきた演出制御コマンドを取り込むコマンド受信バッファが、例えば演出制御ＲＡＭ２０２に用意される。演出制御ＣＰＵ２００は、まず図１３のステップＳ３０１で、コマンド受信バッファの読み出しアドレスを示すリードポインタと、書き込みアドレスを示すライトポインタの確認を行う。
ライトポインタは、コマンド受信に応じて更新され、それに応じてライトポインタで示されるアドレスに、受信した演出制御コマンドが格納されていく。リードポインタは、コマンド受信バッファからの読み出しを行った際に更新（ステップＳ３０２）される。従って、もしリードポインタ＝ライトポインタでなければ、まだ読み出していない演出制御コマンドがコマンド受信バッファに格納されているということになる。そこでリードポインタ＝ライトポインタでなければステップＳ３０２に進み、演出制御ＣＰＵ２００はコマンド受信バッファにおいてリードポインタで示されるアドレスから１バイトのコマンドデータをロードする。この場合、次の読み出し（ロード）のためにリードポインタをインクリメントしておく。

【０１３９】
上述したように１つの演出制御コマンドは、モードとしての１バイトとイベントとしての１バイトの２バイトで形成されている。演出制御ＣＰＵ２００はステップＳ３０３で、ロードした１バイトのコマンドデータが、モードであるかイベントであるかを確認する。この確認は、８ビット（Ｂｉｔ０～Ｂｉｔ７）のうちのＢｉｔ７の値により行われる。
そしてモードであれば、コマンドの上位データ受信の処理として、ステップＳ３０４に進み、読み出したコマンドデータを、レジスタ「コマンドＨＩバイト」にセーブする。また「コマンドＬＯバイト」のレジスタをクリアする。そしてステップＳ３０１に戻る。
続いても、リードポインタ＝ライトポインタでなければ、まだ読み出していない演出制御コマンドがコマンド受信バッファに格納されていることになるため、ステップＳ３０２に進み、演出制御ＣＰＵ２００はコマンド受信バッファにおいてリードポインタで示されるアドレスから１バイトのコマンドデータをロードする。またリードポインタをインクリメントする。

そして読み出したコマンドがイベントであれば、コマンドの下位データ受信の処理として、ステップＳ３０３からＳ３０５に進み、読み出したコマンドデータを、レジスタ「コマンドＬＯバイト」にセーブする。

【０１４０】

モード及びイベントのコマンドデータが、レジスタ「コマンドＨＩバイト」「コマンドＬＯバイト」にセーブされることで、演出制御ＣＰＵ２００は１つのコマンドを取り込んだことになる。

そこで演出制御ＣＰＵ２００はステップＳ３０６で、取り込んだコマンドに応じた処理を行う。具体例は図１４，図１５で後述する。

【０１４１】

リードポインタ＝ライトポインタとなるのは、演出制御ＣＰＵ２００がまだ取り込んでいない演出制御コマンドがコマンド受信バッファには存在しない場合である。そこで、新たな取り込みは不要であるため、ステップＳ３０７に進む。この場合、図柄コマンド待ちの状態で１００ｍｓ経過したか否かを確認する。そのような事態となっていなければ、図１３のコマンド解析処理を終える。一方、図柄コマンド待ちの状態で１００ｍｓ経過していたのであれば、ステップＳ３０８でコマンド欠落における図柄変動の処理を行うこととなる。

【０１４２】

上記のステップＳ３０６におけるコマンド対応処理の例を図１４，図１５で説明する。

図１４Ａは、コマンド対応処理としての基本処理を示している。２バイトの演出制御コマンドの受信に応じて、演出制御ＣＰＵ２００はまず図１４ＡのステップＳ３２１で、現在テストモード中であるか否かを確認する。テストモード中であれば、ステップＳ３２２ですべての演出シナリオのクリア、音出力の停止、ランプ部６３，６４におけるＬＥＤ１２０の消灯を行う。そしてステップＳ３２３でテストモードを終了する。

テストモード中でなければ、これらの処理は行わない。

そして演出制御ＣＰＵ２００は、ステップＳ３３０として、取り込んだ演出制御コマンドについての処理を行うことになる。

【０１４３】

演出制御コマンドとしては、例えば特別図柄変動パターン指定コマンド、特別図柄指定コマンド、保留球減算コマンド、保留球加算コマンド、エラー表示指定コマンド、大当たり開始指定コマンド・・・など多様に設定されている。ステップＳ３３０では、演出制御ＣＰＵ２００は、受信したコマンドに対応して、プログラムで規定された処理を行う。ここでは図１４Ｂ～図１４Ｅ、図１５Ａ～図１５Ｃに７つの演出制御コマンドを挙げて、具体的処理を例示する。

【０１４４】

図１４Ｂは、ステップＳ３３０でのコマンド処理として、変動パターンコマンドを取り込んだ場合の処理Ｓ３３０ａを示している。

この場合、演出制御ＣＰＵ２００は、ステップＳ３３１で、図柄コマンド待ち状態をセットする処理を行う。これは変動パターンコマンドと、図柄コマンドが順次受信されることで、演出制御ＣＰＵ２００が図柄の変動表示の制御を行うためである。

【０１４５】

図１４Ｃは、ステップＳ３３０でのコマンド処理として、図柄指定コマンドを取り込んだ場合の処理Ｓ３３０ｂを示している。

この場合、演出制御ＣＰＵ２００は、まずステップＳ３３２で、変動パターンコマンド受信済みであるか否かを確認する。受信していなければそのまま処理を終える。

図柄指定コマンドを受信した際に、既に変動パターンコマンド受信済みであれば、ステップＳ３３３に進み、まず役物原点補正の動作についてのシナリオ登録を行う。そしてステップＳ３３４で、図柄変動フラグをセットする。図柄変動フラグは、第１特別図柄、第２特別図柄、普通図柄のそれぞれに対応してそれぞれ設けられ、それぞれのフラグで変動状態を表すものとされる。例えば各２ビットの第１特別図柄変動フラグＦＺ１、第２特別

図柄変動フラグＦＺ２、普通図柄変動フラグＦＺ３が用意され、それぞれについて変動中、停止中（当たり）、停止中（外れ）が示される。ここでは変動開始に伴い、対応する図柄変動フラグ（ＦＺ１，ＦＺ２，ＦＺ３のいずれか）を、「変動中」を示す値にセットする。

なお図柄変動フラグは、当たりの場合は、図柄変動終了時に所定時間「停止中（当たり）」を示す値にセットされ、その後、図柄変動が開始されるまで「停止中（外れ）」を示す値にセットされる。

ステップＳ３３５で演出制御ＣＰＵ２００は、変動開始処理を行う。その後、変動開始に応じてステップＳ３３６で変動パターンコマンドの削除を行う。

【０１４６】

図１４Ｄは、ステップＳ３３０でのコマンド処理として、電源投入コマンドを取り込んだ場合の処理Ｓ３３０ｃを示している。

この場合、演出制御ＣＰＵ２００は、ステップＳ３３７で演出制御ＲＡＭ２０２のクリア処理を行う。例えばコマンド受信／送信バッファ、操作部６０についての入力情報バッファの内容や、チェックサムの記憶領域のクリアを行う。

そしてステップＳ３３８でエラー解除コマンドの送信、ステップＳ３３９で役物エラー情報のクリア、ステップＳ３４０で役物動作の停止、ステップＳ３４１で電源投入のシナリオ登録、ステップＳ３４２で液晶制御基板５２へ送信する電源投入コマンドのセットを順次行う。

【０１４７】

図１４Ｅは、ステップＳ３３０でのコマンド処理として、ＲＡＭクリアコマンドを取り込んだ場合の処理Ｓ３３０ｄを示している。

この場合、演出制御ＣＰＵ２００は、ステップＳ３４３で演出制御ＲＡＭ２０２のクリア処理を行う。例えばコマンド受信／送信バッファ、操作部６０についての入力情報バッファの内容や、チェックサムの記憶領域のクリアを行う。

そしてステップＳ３４４でエラー解除コマンドの送信、ステップＳ３４５でＲＡＭクリアエラーセットと、エラー報知タイマのセットを行う。さらにステップＳ３４６で演出制御ＲＡＭ２０２における抽選処理に関する情報のクリア、ステップＳ３４７で、シナリオに関する情報のクリアを行う。そしてステップＳ３４８で液晶制御基板５２へ送信する電源初期投入表示（ＲＡＭクリア）コマンドのセットを行う。

【０１４８】

図１５Ａは、ステップＳ３３０でのコマンド処理として、保留加算コマンドを取り込んだ場合の処理Ｓ３３０ｅを示している。

この場合、演出制御ＣＰＵ２００は、ステップＳ３５０で第１特別図柄変動表示ゲーム又は第２特別図柄変動表示ゲームのいずれか対応する方の保留数レジスタ（演出制御ＲＡＭ２０２内で設定された第１保留数又は第２保留数としてのレジスタ領域）における保留数の値を読み出して、加算する処理を行う。そしてステップＳ３５１で液晶制御基板５２に対して保留数表示のためのコマンドをセットし、ステップＳ３５２で第１保留数又は第２保留数として更新した値をセットする。

【０１４９】

図１５Ｂは、ステップＳ３３０でのコマンド処理として、保留加算コマンドを取り込んだ場合の処理Ｓ３３０ｆを示している。

この場合、演出制御ＣＰＵ２００は、ステップＳ３６０で第１特別図柄変動表示ゲーム又は第２特別図柄変動表示ゲームのいずれか対応する方の保留数レジスタ（演出制御ＲＡＭ２０２内で設定された第１保留数又は第２保留数としてのレジスタ領域）における保留数の値を読み出して、減算する処理を行う。そしてステップＳ３６１で液晶制御基板５２に対して保留数表示のためのコマンドをセットし、ステップＳ３６２で第１保留数又は第２保留数として更新した値をセットする。

【０１５０】

図１５Ｃは、ステップＳ３３０でのコマンド処理として、状態報知コマンドを取り込ん

だ場合の処理Ｓ３３０ｇを示している。

この場合、演出制御ＣＰＵ２００は、ステップＳ３７１で遊技状態報知フラグをセットする。遊技状態報知フラグとしては、例えば「大当たり中」「確変中」「時短中」を示すフラグＦＪ１，ＦＪ２，ＦＪ３が用意される。演出制御ＣＰＵ２００はステップＳ３７１では状態報知コマンドで指示される遊技状態に応じて、いずれかの対応する遊技状態報知フラグをセットすることになる。

【０１５１】

（1-6-4：シナリオ登録・削除処理）

次にシナリオ登録・削除処理について説明する。シナリオとは演出制御やエラー処理その他、各種の実行すべき動作を規定したデータである。実行すべきシナリオのデータは、シナリオ登録情報として演出制御ＲＡＭ２０２のワーク領域に登録される。図２０Ａに示すシナリオ登録情報の構造については後述するが、シナリオ登録情報としては、０～６３までの６４個のシナリオチャネルが用意されている。この６４個のシナリオチャネルに登録されたシナリオは同時に実行可能とされる。以下、シナリオチャネルを「ｓＣＨ」で示す。

シナリオ登録処理とは、シナリオ登録情報における任意のシナリオチャネルに、登録を要求されたシナリオ番号のシナリオを登録する処理である。原則的には、ｓＣＨ０～ｓＣＨ６３のシナリオチャネルは、どのチャネルが用いられても良い。

【０１５２】

以下説明するシナリオ登録は、例えば図１１のステップＳ１０６のエラー処理、ステップＳ１０７のデモ処理、ステップＳ１０８のコマンド解析処理などの処理過程において、必要時に呼び出され、実行される。例えば先に述べた図１４ＣのステップＳ３３３、図１４ＤのステップＳ３４１などが、シナリオ登録が実行される場合の一例となる。

【０１５３】

図１６の処理として、演出制御ＣＰＵ２００は、コマンド或いはプログラム上で指定されるシナリオに対応したシナリオ番号と待機時間（delay）の値を、ワーク（シナリオ登録情報）に登録する処理を行う。

ステップＳ４０１で演出制御ＣＰＵ２００は、まず今回登録すべきシナリオ番号が正常であるか否かを確認する。シナリオ番号があり得ない番号の場合は、何もせずに処理を終える。

【０１５４】

シナリオ番号が適正であれば、演出制御ＣＰＵ２００はステップＳ４０２で変数Ｂをゼロに設定する。変数Ｂは、ｓＣＨ０～ｓＣＨ６３のシナリオチャネルのうちで、空きチャネルを順次探索するために使用する変数である。さらに変数Ｂは、まだ探索（空きチャネルであるか否かの確認）をしていないシナリオチャネルが残っているか否かを判断するための変数を兼ねている。

【０１５５】

演出制御ＣＰＵ２００は、ステップＳ４０３で、「追加ポインタ」＋Ｂのシナリオチャネルが空きであるか否かを確認する。追加ポインタとは、シナリオ登録を行った際に、後述のステップＳ４０７で設定されるポインタである。図１６の処理を開始する時点では、追加ポインタの値は、前回登録したシナリオチャネルの次のシナリオチャネルを示す値となっている。なお、追加ポインタの初期値（初期状態から図１６の処理が初めて行われる時の値）は０である。

「追加ポインタ」＋Ｂのシナリオチャネルが空きでなければ、演出制御ＣＰＵ２００はステップＳ４０４で変数Ｂをインクリメントする。

ステップＳ４０５でＢ＜６４でなければ、この図１６の処理を終える。これは全シナリオチャネルについて探索を行ったが、空きチャネルがなくてシナリオ登録が不可能となる場合である。

まだ全シナリオチャネルの探索（空きチャネルであるか否かの確認）を行っていない時

点では、ステップＳ４０５でＢ＜６４である。その場合はステップＳ４０６の確認処理を行い、「追加ポインタ」＋Ｂの値がシナリオチャネルｓＣＨの最大値を超えた値「６４」以上となっていなければステップＳ４０３に戻る。
また「追加ポインタ」＋Ｂの値が「６４」以上となっていた場合は、ステップＳ４０７で追加ポインタの値の補正処理を行う。具体的には「追加ポインタ」＋Ｂの値が「６４」となった場合、追加ポインタの値を「６４」減算する処理を行う。そしてステップＳ４０３に戻る。
【０１５６】
このステップＳ４０３〜Ｓ４０７の処理によれば、前回登録したシナリオチャネルの次のシナリオチャネルから、順次シナリオチャネルが空きか否かが確認されることになる。
つまり、ステップＳ４０４での変数Ｂのインクリメントにより、ステップＳ４０３が行われるたびに確認されるシナリオチャネルｓＣＨが、順次１つずつ進行することとなる。
また、変数ＢはステップＳ４０２でゼロリセットされてからステップＳ４０４でインクリメントされるものであるため、ステップＳ４０５でＢ＜６４とはならない場合（つまり変数Ｂが６４に達した場合）は、既にステップＳ４０３の処理が６４回行われた場合である。これは全シナリオチャネルｓＣＨ０〜ｓＣＨ６３を調べたが、空きがなかったと判断された場合である。そのため登録不可能として図１６の処理を終えることとなる。
【０１５７】
また、「追加ポインタ」の値は、前回登録したシナリオチャネルの次のシナリオチャネルを示す値であるため、まだ全シナリオチャネルの空き確認を行っていないＢ＜６４の時点でも、「追加ポインタ」＋Ｂの値（つまり次に確認しようとするシナリオチャネルｓＣＨの番号）が「６４」以上となることがある。具体的には、直前のステップＳ４０３の時点で「追加ポインタ」＋Ｂの値が６３であり、シナリオチャネルｓＣＨ６３について確認した後、ステップＳ４０４で変数Ｂがインクリメントされた場合である。このままでは、次に存在しないシナリオチャネルｓＣＨ６４を指定することとなる。
そこでステップＳ４０６でこの点を確認し、「追加ポインタ」＋Ｂの値が「６４」となっていたら、ステップＳ４０７では、次に確認するシナリオチャネルを「ｓＣＨ０」に戻すために、追加ポインタの補正を行う。つまり追加ポインタの値を－６４することで、「追加ポインタ」＋Ｂの値が「０」となるようにする。これにより次のステップＳ４０３では、シナリオチャネルｓＣＨ０が空きであるか否か確認されるようにする。
【０１５８】
ある時点のステップＳ４０３の処理で、空きのシナリオチャネルが発見されたら、演出制御ＣＰＵ２００はステップＳ４０８に進み、その空きのシナリオチャネルに、シナリオ番号、及び待機時間（delay）をセットする。またその他のシナリオ管理に必要なデータをゼロクリアする。
そしてステップＳ４０９で、追加ポインタを、登録を行ったシナリオチャネル＋１の値に更新する。つまり今回登録を行ったシナリオチャネルの次のシナリオチャネルの値を、追加ポインタとして記憶しておき、次回の登録処理に使用できるようにする。なお、本実施形態ではシナリオチャネルはｓＣＨ０〜ｓＣＨ６３の６４チャネルのため追加ポインタの最大値は６３となる。
【０１５９】
この図１６の処理によれば、シナリオ登録の際に、前回登録を行ったシナリオチャネルの次のシナリオチャネルから、空きチャネルが探索される。追加ポインタの初期値は「０」であり、その後、登録に応じてステップＳ４０７で更新されていくが、この処理によれば、多くの場合、シナリオチャネルｓＣＨ０から順に使用されてシナリオ登録が行われる。そして、シナリオ登録の際には、前回の登録チャネルの次のシナリオチャネルから空きの確認が行われる。従って、殆どの場合、素早く空きチャネルが発見でき、シナリオ登録処理を効率的に実行することができる。これにより演出制御ＣＰＵ２００の処理負担は軽減される。
【０１６０】

なお、処理の変形例として、ステップＳ４０７で更新する追加ポインタの値は、登録を行ったシナリオチャネルの番号としておき、ステップＳ４０２では変数Ｂに１を代入してもよい。但しその場合、ステップＳ４０５ではＢ≦６５であるか否かの判断を行うようにする。

【０１６１】

次にシナリオ削除処理について説明する。これはワークの或るシナリオチャネルに登録されているシナリオを削除する処理である。

図１７Ａは、或るシナリオをシナリオ登録情報から削除する場合の演出制御ＣＰＵ２００の処理を示している。

演出制御ＣＰＵ２００は、コマンド或いはプログラムグラム上で指定される、削除するシナリオ番号（後述のメインシナリオ番号（mcNo））の値に基づいて、図１７Ａの処理を開始する。まずステップＳ４２１で演出制御ＣＰＵ２００は、削除要求にかかるシナリオ番号が正常であるか否かを確認する。シナリオ番号があり得ない番号の場合は、削除せずに処理を終える。

【０１６２】

シナリオ番号が適正であれば、演出制御ＣＰＵ２００はステップＳ４２２で変数Ｂをゼロに設定する。この場合の変数Ｂは、ｓＣＨ０～ｓＣＨ６３のシナリオチャネルのうちで、削除対象のシナリオが登録されたチャネルを探索するために使用する変数となる。さらに変数Ｂは、まだ探索（削除対象のシナリオが登録されているか否かの確認）をしていないシナリオチャネルが残っているか否かを判断するための変数を兼ねている。

【０１６３】

演出制御ＣＰＵ２００は、ステップＳ４２３で、Ｂ領域、つまりシナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオが削除対象のシナリオ番号のものであるか否かを確認する。シナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオが削除対象のシナリオ番号でなければ、ステップＳ４２４で変数Ｂをインクリメントし、ステップＳ４２５でＢ＜６４であることを確認して、ステップＳ４２３の処理を行う。

このステップＳ４２３，Ｆ４２４の処理によれば、シナリオチャネルｓＣＨ０からシナリオチャネルｓＣＨ６３に向かって順に、削除対象のシナリオを探索していくこととなる。

ステップＳ４２３でシナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオが削除対象のシナリオ番号であった場合は、ステップＳ４２６に進み、Ｂ領域、つまりシナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオを削除する処理を行う。

以上により、要求された或るシナリオ番号のシナリオをワーク（シナリオ登録情報）から削除する処理が行われる。

なお、ステップＳ４２５でＢ＜６４ではないと判断される場合、つまり変数Ｂが６４に達した場合は、シナリオチャネルｓＣＨ０～ｓＣＨ６３の全てを探索したが、削除対象のシナリオが登録されていなかったということになるため、処理を終える。

【０１６４】

図１７Ｂは、或る範囲のシナリオを削除する処理を示している。削除シナリオが範囲で指定された場合に、この処理が行われる。

演出制御ＣＰＵ２００は、コマンド或いはプログラム上で、或るシナリオ番号の範囲で削除指定された場合、まずステップＳ４３１で変数Ｂをゼロに設定する。この場合の変数Ｂは、ｓＣＨ０～ｓＣＨ６３のシナリオチャネルのうちで、削除対象範囲に該当するシナリオが登録されたチャネルを探索するために使用する変数となる。さらに変数Ｂは、まだ探索（削除対象範囲に該当するシナリオが登録されているか否かの確認）をしていないシナリオチャネルが残っているか否かを判断するための変数を兼ねている。

【０１６５】

演出制御ＣＰＵ２００は、ステップＳ４３２で、Ｂ領域、つまりシナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオが、削除対象とされた範囲内のシナリオ番号であるか否かを確認する。そしてシナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオが削除

対象の範囲内のシナリオ番号であれば、ステップＳ４３３でシナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオを削除する。そしてステップＳ４３４に進む。
シナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオが、削除対象とされた範囲内のシナリオ番号ではなければ、ステップＳ４３３を行わずにステップＳ４３４に進む。
演出制御ＣＰＵ２００はステップＳ４３４では、変数Ｂをインクリメントし、ステップＳ４３５でＢ＜６４であることを確認して、ステップＳ４３２に戻る。変数Ｂが６４に達していたら、全シナリオチャネルｓＣＨ０～ｓＣＨ６３について処理を完了したことになるため、このシナリオ範囲削除処理を終える。
以上により、シナリオ番号範囲の１又は複数のシナリオについて、ワーク（シナリオ登録情報）からの削除が行われる。

【０１６６】
図１７Ｃは登録されているシナリオを全て削除する処理を示している。例えばシステム上の都合により、やむを得ずシナリオを削除する際に呼び出される処理である。なお、保護対象とされたシナリオは削除しないようにする。
演出制御ＣＰＵ２００は、シナリオ全削除が要求された場合、まずステップＳ４４１で変数Ｂをゼロに設定する。この場合の変数Ｂは、ｓＣＨ０～ｓＣＨ６３のシナリオチャネルのうちで、保護対象のシナリオを登録したシナリオチャネルを確認するために使用する変数となる。さらに変数Ｂは、まだ確認（保護対象のシナリオが登録されているか否かの確認）をしていないシナリオチャネルが残っているか否かを判断するための変数を兼ねている。

【０１６７】
演出制御ＣＰＵ２００は、ステップＳ４４２で、Ｂ領域、つまりシナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオが、保護対象のシナリオであるか否かを確認する。そして保護対象のシナリオであれば、削除せず、一方、保護対象のシナリオでなければ、ステップＳ４４３で、そのシナリオチャネルｓＣＨ（Ｂ）に登録されているシナリオを削除する。
そしてステップＳ４４４で変数Ｂをインクリメントし、ステップＳ４４５でＢ＜６４であればステップＳ４４２に戻る。変数Ｂが６４に達していたら、全シナリオチャネルｓＣＨ０～ｓＣＨ６３について処理を完了したことになるため、このシナリオ全削除処理を終える。
以上により、ワークにシナリオ登録情報として登録されているシナリオについて、保護対象のシナリオを除く全シナリオの削除が行われる。

【０１６８】
（1-6-5：シナリオ更新処理）

続いてメイン処理の１６ｍｓ処理のステップＳ１１０で行われるシナリオ更新処理について説明する。シナリオ更新処理では図１８、図１９で説明するようにメインシナリオとサブシナリオの更新が行われる。

【０１６９】
まずシナリオ登録情報の構造を図２０、図２１で説明する。図２０Ａは、メインシナリオ及びサブシナリオとしてのシナリオ登録情報の構造を示している。このシナリオ登録情報は演出制御ＲＡＭ２０２のワークエリアを用いて設定される。
本実施形態では、先にも述べたようにシナリオ登録情報は、シナリオチャネルｓＣＨ０～ｓＣＨ６３の６４個のチャネルを有するものとされる。各シナリオチャネルｓＣＨに登録されたシナリオについては同時に実行可能とされる。

【０１７０】
図示のように各シナリオチャネルｓＣＨに登録できる情報としては、サブシナリオ更新処理で用いるサブシナリオタイマ（scTm）と前回時間（scPrevTm）、音／モータサブシナリオテーブルの実行ラインを示すサブシナリオ実行ライン（scIx）、ランプサブシナリオテーブルの実行ラインを示すサブシナリオ実行ラインｌｍｐ（lmpIx）、シナリオ更新処

理に用いるメインシナリオタイマ（msTm）、メインシナリオテーブルの実行ラインを示すメインシナリオ実行ライン（mcIx）、メインシナリオ番号（mcNo）、メインシナリオに付加可能なオプションデータであるメインシナリオオプション（mcOpt）、ユーザオプション（userFn）、待機時間（delay）、チェックサム（checkSum）がある。

【０１７１】

スピーカ部５９による音出力、ランプ部６３，６４による発光、及び可動体役物モータ６５による可動体役物の駆動による演出を開始するときには、待機時間（delay）とメインシナリオ番号（mcNo）をシナリオチャネルｓＣＨ０～ｓＣＨ６３のうちの空いているシナリオチャネルに登録する。

待機時間（delay）は、シナリオチャネルｓＣＨに登録してからそのシナリオが開始されるまでの時間を示す。なおこの待機時間（delay）は、第一実施形態では１６ｍｓ処理毎に１減算される。待機時間（delay）が０の場合に、登録されたデータに対応した処理が実行されることとなる。

【０１７２】

図２２には、メインシナリオテーブルの一部として、シナリオ番号１，２，３の例を示している。各シナリオ番号のシナリオとしては、シナリオの各ライン（行）に時間データとしてメインシナリオタイマ（msTm）の値が記述されるとともに、サブシナリオ番号（scNo）、オプション（OPT）を記述することができる。即ちメインシナリオテーブルでは、メインシナリオタイマ（msTm）による時間として、実行されるべきサブシナリオ（及び場合によってはオプション）が指定される。またシナリオ最終行には、シナリオデータ終了コードD_SEEND、又はシナリオデータループコードD_SELOPが記述される。

なお、第一実施形態では、メインシナリオタイマ（msTm）の値は、メインシナリオの開始時から１６ｍｓ処理で行われるシナリオ更新の処理で＋１されるため、「１」とは１６ｍｓを示すものとなる。各シナリオ番号のシナリオテーブルは、或る行におけるメインシナリオタイマ（msTm）の時間を経過すると、次の行へ進むことになる。各行の時間データは、その行が終わるタイミングを示している。

例えばシナリオ番号２の場合、１５００×１６ｍｓの時間としてサブシナリオ番号２の動作が指定され、次の５００×１６ｍｓの時間としてサブシナリオ番号２０の動作が指定され、次の２０００×１６ｍｓの時間としてサブシナリオ番号２１の動作が指定されている。その次の行はシナリオデータ終了コードD_SEENDである。シナリオデータ終了コードD_SEENDの場合、シナリオ登録情報（ワーク）から、このシナリオが削除される（後述する図１８のステップＳ６１７参照）。

【０１７３】

次に図２０Ｂでランプデータ登録情報の構造を説明する。ランプデータ登録情報としては、ランプサブシナリオテーブルから選択されたシナリオ、即ちランプ部６３，６４による演出動作（点灯パターン）を示す情報が登録される。このランプデータ登録情報も演出制御ＲＡＭ２０２のワークエリアを用いて設定される。

本実施形態では、ランプデータ登録情報は、ランプチャネルｄｗＣＨ０～ｄｗＣＨ１５の１６個のチャネルを有するものとされる。各ランプチャネルｄｗＣＨ０～ｄｗＣＨ１５には優先順位が設定されており、ランプチャネルｄｗＣＨ０からｄｗＣＨ１５に向かって順にプライオリティが高くなる。従ってランプチャネルｄｗＣＨ１５に登録されたシナリオ（ランプサブシナリオ）が最も優先的に実行される。また例えばランプチャネルｄｗＣＨ３、ｄｗＣＨ１０にシナリオが登録されていれば、ランプチャネルｄｗＣＨ１０に登録されたシナリオが優先実行される。

なお、ランプチャネルｄｗＣＨ０は主にＢＧＭ（Back Ground Music）に付随するランプ演出、ランプチャネルｄｗＣＨ１５はエラー関係のランプ演出に用いられ、ランプチャネルｄｗＣＨ１～ｄｗＣＨ１４が通常演出に用いられる。

【０１７４】

各ランプチャネルｄｗＣＨに登録できる情報としては、図示のように、登録した点灯パターンの番号を示す登録点灯ナンバ（lmpNew）、実行する点灯パターンの番号を示す実行

点灯ナンバ（lmpNo）、ランプサブシナリオの実行ラインを示すオフセット（offset）、実行時間（time）、チェックサム（checkSum）がある。

【０１７５】

図２３Ａにランプサブシナリオテーブルの一部として、ランプサブシナリオ番号１，２，３の例を示している。各番号のランプサブシナリオとしては、シナリオの各ライン（行）に時間データ（time）の値が記述されるとともに、ランプチャネルと、各種の点灯パターンを示すランプナンバが記述される。また最終行には、ランプシナリオデータ終了コードD_LSENDが記述される。

【０１７６】

このランプサブシナリオテーブルにおいて、各ラインの時間データ（time）は、そのサブシナリオが開始されてからの、当該ラインが開始される時間を示している。

１６ｍｓ毎にメインシナリオタイマ（msTm）と、テーブルの時間データを比較して、一致した場合に、そのラインのランプナンバが、図２０Ｂのランプデータ登録情報に登録される。登録されるランプチャネルｄｗＣＨは、当該ラインに示されたチャネルとなる。

例えば、上述の或るシナリオチャネルｓＣＨにおいて、図２２に示したシナリオ番号２が登録され、サブシナリオ番号２が参照されるとする。図２３Ａに示したランプサブシナリオ番号２では、１ライン目に時間データ（time）＝０としてランプチャネル５（ｄｗＣＨ５）及びランプナンバ５が記述されている。この場合、メインシナリオタイマ（msTm）＝０の時点で、まず当該１ライン目の情報が図２０Ｂのランプデータ登録情報のランプチャネルｄｗＣＨ５に、登録点灯ナンバ（lmpNew）＝５として登録される。シナリオ登録情報のサブシナリオ実行ラインｌｍｐ（lmpIx）の値は、次のラインの値（２ライン目）に更新される。これはランプチャネルｄｗＣＨ５という比較的低い優先度で、点灯ナンバ５の点灯パターン動作の実行を行うための登録となる。

２ライン目については、５００×１６ｍｓとなった時点で同様の処理が行われる。即ちランプデータ登録情報のランプチャネルｄｗＣＨ５に、登録点灯ナンバ（lmpNew）＝６（つまり点灯ナンバ６の点灯パターンの指示）が登録される。

なお、時間データ（time）が連続する２ラインで同一の値であったら、その各ラインについての処理は同時に開始されることとなる。

後述するＬＥＤ駆動データ更新処理では、このように更新されるランプデータ登録情報に基づいて、ＬＥＤ駆動データが作成される。

【０１７７】

次に図２０Ｃでモータデータ登録情報の構造を説明する。モータデータ登録情報としては、音／モータサブシナリオテーブルから選択されたシナリオを示す情報が登録される。このモータデータ登録情報も演出制御ＲＡＭ２０２のワークエリアを用いて設定される。

本実施形態では、モータデータ登録情報は、モータチャネルｍＣＨ０～ｍＣＨ７の８個のチャネルを有するものとされる。

各モータチャネルｍＣＨに登録できる情報としては、図示のように、実行動作ナンバ（no）、登録動作ナンバ（noNew）、動作カウント（lcnt）、励磁カウンタ（tcnt）、実行ステップ（step）、動作ライン（offset）、親（移行元）／子（移行先）の属性（attribute）、親ナンバ（retNo）、戻りアドレス（retAddr）、ループ開始ポイント（roopAddr）、ループ回数（roopCnt）、エラーカウンタ（errCnt）、現在の入力情報（currentSw）、ソフト上のスイッチ情報（softSw）、ソフト上のカウント（softCnt）がある。

【０１７８】

また図２１は、音データ登録情報を示している。音データ登録情報としては、音／モータサブシナリオテーブルから選択されたシナリオを示す情報が登録される。この音データ登録情報も演出制御ＲＡＭ２０２のワークエリアを用いて設定される。

本実施形態では、音データ登録情報は、音チャネルａＣＨ０～ａＣＨ１５の１６個のチャネルを有するものとされる。

各音チャネルａＣＨに登録できる情報としては、図示のように、ボリューム遷移量（frzVq）、ボリューム（frzVl）、遷移量変化（rsv2）、ボリューム変化（rsv1）、フレーズ

変化（rsv0）、ステレオ（frzSt）、ループ（frzLp）、フレーズ番号ｈｉ（frzHi）、フレーズ番号ｌｏｗ（frzLo）がある。

【０１７９】

図２３Ｂに音／モータサブシナリオテーブルの一部として、音／モータサブシナリオ番号１，２の例を示している。各番号の音／モータサブシナリオとしては、シナリオの各ライン（行）に時間データ（time）の値（ｍｓ）が記述されるとともに、ＢＧＭ、予告音、エラー音、音コントロール、モータ、ソレノイド／ユーザオプションの情報が記述される。また最終行には、シナリオデータ終了コードD_SEENDが記述される。

この音／モータサブシナリオテーブルに関しては、サブシナリオタイマ（scTm）が０になったら（なお最初は０である）、この音／モータサブシナリオテーブルの時間データ（time）の値をサブシナリオタイマ（scTm）にセットする。なお、各ラインの時間データ（time）は、当該ラインが終了するタイミングを示している。サブシナリオタイマ（scTm）には、絶対時間を記述するが、従って、セットする時間データ値は、（当該ラインの時間データ）－（前回ラインの時間データ）の値である。

当該ラインのＢＧＭのデータは、ＢＧＭのフレーズ番号やボリューム値等の音データ登録情報に登録する情報で構成され、音データ登録情報における音チャネルａＣＨ０（ステレオの場合は加えてａＣＨ１）にセットされる。

当該ラインの予告音のデータは、予告音のフレーズ番号やボリューム値等の音データ登録情報に登録する情報で構成され、音チャネルａＣＨ２～ａＣＨ１４の空いているところにセットされる。

当該ラインのエラー音のデータは、エラー音のフレーズ番号やボリューム値等の音データ登録情報に登録する情報で構成され、音チャネルａＣＨ１５にセットされる。

音コントロールのデータは、下位６バイトでチャネル情報、上位２バイトでコントロール情報とされている。

モータのデータは、モータ１個につき１バイトでモータの動作パターン番号を示すように構成されている。モータ番号に対応するモータチャネルに動作パターン番号がセットされる。

【０１８０】

図１２のステップＳ２０３として説明したモータ動作更新処理では、図１８，図１９のシナリオ更新処理で更新されるモータデータ登録情報に基づいて、モータ駆動データが作成される。

また図１１のステップＳ１１１の音再生処理では、図１８，図１９のシナリオ更新処理で更新される音データ登録情報に基づいて、再生出力が行われる。

【０１８１】

以上の各情報を用いたシナリオ更新処理について、図１８，図１９で説明する。

図１１の１６ｍｓ処理のステップＳ１１０として実行される図１８のシナリオ更新処理では、演出制御ＣＰＵ２００はループ処理ＬＰ１として、シナリオチャネルｓＣＨ０～ｓＣＨ６３のそれぞれについて、ステップＳ６０１～Ｓ６１６の処理を行う。当該ループ処理の各回の処理対象のシナリオチャネルを「ｓＣＨｎ」として説明する。

【０１８２】

演出制御ＣＰＵ２００は、ステップＳ６０１でシナリオチャネルｓＣＨｎの待機時間（delay）を確認する。待機時間（delay）＝０でなければステップＳ６０２で待機時間（delay）の値を－１（デクリメント）する。そしてステップＳ６１５でシナリオチャネルｓＣＨｎ関連のデータのチェックサムを算出し、保存する。またステップＳ６１６でシナリオチャネルｓＣＨｎ関連のデータをバックアップ保存する。これでシナリオチャネルｓＣＨｎについての１回の処理を終える。

【０１８３】

一方、ステップＳ６０１で待機時間（delay）＝０であることが確認された場合は、演出制御ＣＰＵ２００はステップＳ６０３に進み、シナリオチャネルｓＣＨｎに登録されているメインシナリオ番号（mcNo）と、メインシナリオ実行ライン（mcIx）に対応するメイ

ンシナリオテーブルのアドレスを特定する。
ステップＳ６０４では、当該特定したアドレスで示される、メインシナリオテーブルの或るメインシナリオ番号の実行ラインが、終了コードD_SEEND（図２２参照）が記述されている最終ラインであるか否かを確認する。
終了コードが記述された最終ラインであった場合は、当該シナリオチャネルｓＣＨｎに登録されたシナリオで実行すべき処理は終了したことになるため、ステップＳ６１７で、そのシナリオチャネルｓＣＨｎに登録されているシナリオをシナリオ登録情報（ワーク）から削除する。
なお、この場合のシナリオ登録の削除は、シナリオ終了に応じた通常の削除である。先に図１７で説明したシナリオ削除は、この通常削除以外の、例えば未終了のシナリオをシナリオ登録情報（ワーク）から削除する処理であることを付言しておく。
【０１８４】
ステップＳ６０４で、メインシナリオテーブルの当該ラインが終了コードではない場合は、演出制御ＣＰＵ２００はステップＳ６０５に進み、まず当該シナリオチャネルｓＣＨｎのメインシナリオオプション（mcOpt）に、当該ラインのオプション（OPT）をセットする。次にステップＳ６０６で、シナリオ番号が０であるか否かを確認し、０でなければステップＳ６０７で、ユーザオプション（userFn）に０を代入する。
そしてステップＳ６０８で、当該シナリオチャネルｓＣＨｎで指定されるサブシナリオの更新を行う。サブシナリオの更新については図１９で後述する。
【０１８５】
ステップＳ６０９では、演出制御ＣＰＵ２００はメインシナリオタイマ（msTm）を＋１（インクリメント）する。そしてステップＳ６１０では、メインシナリオタイマ（msTm）の値と、メインシナリオテーブルの該当ラインの時間データ（図２２参照：msTmで記述）を比較する。先に述べたように、メインシナリオテーブルの各ラインの時間データは、そのラインの終了タイミングを規定する。従ってメインシナリオタイマ（msTm）の値がメインシナリオテーブルの当該ラインに記述された時間データ以上であれば、そのラインの処理は終了し、次のラインを対象とする。その場合ステップＳ６１０からＳ６１１に進み、シナリオチャネルｓＣＨｎのメインシナリオ実行ライン（mcIx）を＋１する。つまり次回は、次のラインが対象となるようにする。
またその場合、ステップＳ６１２で次のラインがループ指定であるか否かを確認する。図２２のメインシナリオテーブルにおいてシナリオ番号１では、最終ラインがシナリオデータループコードD_SELOPとされている例を示したが、このようにループ指定されていた場合は、ステップＳ６１３で、メインシナリオ実行ライン（mcIx）にループ行をセットする。
【０１８６】
演出制御ＣＰＵ２００は、ステップＳ６１４では、１つのラインの終了に応じたクリア処理を行う。即ちメインシナリオタイマ（msTm）、サブシナリオタイマ（scTm）、前回時間（scPrevTm）、サブシナリオ実行ライン（scIx）、サブシナリオ実行ラインｌｍｐ（lmpIx）、メインシナリオオプション（mcOpt）、ユーザオプション（userFn）、待機時間（delay）をクリアする。
そしてステップＳ６１５のチェックサム処理，ステップＳ６１６のバックアップ処理を行ってシナリオチャネルｓＣＨｎについての１回の処理を終える。
シナリオ更新処理としては、ループ処理ＬＰ１として、シナリオチャネルｓＣＨ０～ｓＣＨ６３のそれぞれについて以上の処理が実行されることになる。
【０１８７】
ステップＳ６０８で行われるサブシナリオ更新処理を図１９Ａに詳細に示す。
メインシナリオタイマ（msTm）、シナリオチャネルｓＣＨｎと、メインシナリオテーブルに記述されたサブシナリオ番号（scNo）に基づいて、図１９Ａのサブシナリオの更新処理が行われる。
【０１８８】

まずステップＳ６２１で演出制御ＣＰＵ２００は、シナリオチャネルｓＣＨｎが０～６３のいずれかを示しているか否か、つまり適正値であるか否かを確認する。シナリオチャネルｓＣＨｎが６４以上であれば、更新処理不能として図１９Ａの処理を終える。

シナリオチャネルｓＣＨｎが適正値であれば、演出制御ＣＰＵ２００はステップＳ６２２で、サブシナリオ番号（scNo）とサブシナリオ実行ラインｌｍｐ（lmpIx）に対応するランプサブシナリオテーブル（図２３Ａ参照）のアドレスを特定する。

【０１８９】

演出制御ＣＰＵ２００はステップＳ６２３では、ランプデータ登録情報の更新のため、まずランプチャネルｄｗＣＨに０をセットする。つまりまずランプチャネルｄｗＣＨ０を指定した状態とする。

そしてステップＳ６２４で、メインシナリオタイマ（msTm）とランプサブシナリオテーブルの時間データ（time）を比較する。ランプサブシナリオテーブルの時間データ（time）は、当該ライン（サブシナリオ実行ラインｌｍｐ（lmpIx）で示されるライン）が開始される時間（ｍｓ）を示している。従って、メインシナリオタイマの時間（実際にはmsTm×１６ｍｓの時間）が、時間データ（time）以上となっていたら、そのラインについての処理を行う。その場合、ステップＳ６２５でランプチャネルｄｗＣＨの値が正常（ｄｗＣＨ０～ｄｗＣＨ１５の範囲内）であるか否かを確認する。異常な値であれば処理を終える。正常な値であればステップＳ６２６で、現在のラインが、ランプシナリオデータ終了コードD_LSENDが記述されたラインであるか否かを確認する。

【０１９０】

ランプシナリオデータ終了コードD_LSENDが記述されたラインではなければ、演出制御ＣＰＵ２００はステップＳ６２７で、当該ラインに記述されているランプチャネルｄｗＣＨ及びランプナンバを取得し、ステップＳ６２８で、取得したランプチャネルｄｗＣＨに点灯パターンナンバの登録を行う。

点灯パターンナンバの登録処理を図１９Ｂに示している。この場合、まず演出制御ＣＰＵ２００はステップＳ６５１で、当該ラインに記述されているランプチャネルｄｗＣＨの値が正常値であるか否かを判別する。正常値でなければ登録を行わずに処理を終える。

正常値であれば、ステップＳ６５２で、当該ラインに記述されているランプナンバが正常値であるか否かを判別する。正常値でなければ登録を行わずに処理を終える。

ランプチャネルｄｗＣＨ及びランプ番号のいずれもが正常値であれば、ステップＳ６５３でワークのランプデータ登録情報における、ランプチャネルｄｗＣＨに対応する領域に登録点灯ナンバ（lmpNew）と実行点灯ナンバ（lmpNo）をセットする。即ちランプサブシナリオテーブルの当該ラインから取得したランプナンバを、登録点灯ナンバ（lmpNew）にセットし、「０」を実行点灯ナンバ（lmpNo）にセットする。

以上の図１９Ｂの処理をステップＳ６２８で行ったら、演出制御ＣＰＵ２００はステップＳ６２９でサブシナリオ実行ラインｌｍｐ（lmpIx）の値を＋１し、ステップＳ６３０でランプチャネルｄｗＣＨの値を＋１してステップＳ６２４に戻る。

【０１９１】

ステップＳ６２４で、メインシナリオタイマの時間が、サブシナリオ実行ラインｌｍｐ（lmpIx）で示されるラインの時間データ（time）に達していない場合、及びステップＳ６２６でランプシナリオデータ終了コードD_LSENDが確認された場合は、演出制御ＣＰＵ２００の処理はステップＳ６３１へ進む。

【０１９２】

ステップＳ６３１ではサブシナリオタイマ（scTm）が０であるか否かを確認する。０でなければ、音及びモータの登録を行わずにステップＳ６３９へ進む。この場合ステップＳ６４０でサブシナリオタイマ（scTm）の値をデクリメントして、この図１９Ａの処理を終えることとなる。

【０１９３】

ステップＳ６３１の時点でサブシナリオタイマ（scTm）＝０であれば、演出制御ＣＰＵ２００は処理をステップＳ６３２に進め、メインシナリオテーブルに記述されているサブ

シナリオ番号と、現在処理中のシナリオチャネルｓＣＨｎに登録されているサブシナリオ実行ライン（scIx）に対応する音／モータサブシナリオテーブルのアドレスを特定する。
そしてステップＳ６３３では演出制御ＣＰＵ２００は、該当する音／モータサブシナリオ番号のテーブルの、サブシナリオ実行ライン（scIx）で示されるラインが、シナリオデータ終了コードD_SEENDが記述された行であるか否かを確認し、シナリオデータ終了コードD_SEENDが記述された行であれば処理を終了する。
シナリオデータ終了コードD_SEENDが記述された行でなければ、演出制御ＣＰＵ２００はステップＳ６３４で、サブシナリオタイマ（scTm）に、当該ラインの時間データ（time）から前回時間（scPrevTm）を減算した値を代入する。また前回時間（scPrevTm）には、当該ラインの時間データ（time）を代入する。
【０１９４】
そして演出制御ＣＰＵ２００はステップＳ６３５で音の登録を行い、またステップＳ６３６でモータの登録を行う。詳述は避けるが、図２３Ｂに例示した音／モータサブシナリオテーブルの該当サブシナリオ番号の該当ラインの情報を、図２０Ｃのモータデータ登録情報の任意のモータチャネル、及び図２１の音データ登録情報の任意の音チャネルに登録する処理を行う。またステップＳ６３７でそのラインに記述されたソレノイド／ユーザオプションの情報も、シナリオ登録情報等に登録する。
そしてステップＳ６３８で、サブシナリオ実行ライン（scIx）の値として次のラインの値をセットする。そしてステップＳ６３９でサブシナリオタイマ（scTm）が０でなければ、ステップＳ６４０でサブシナリオタイマ（scTm）をデクリメントして処理を終える。
【０１９５】
図１１のステップＳ１１０では、以上のように図１８，図１９の処理が行われ、ワーク上のシナリオ登録情報、ランプデータ登録情報、モータデータ登録情報、音データ登録情報が逐次更新される。これに応じて演出制御が行われることで、シナリオに沿った演出動作が実行される。
【０１９６】
（1-6-6：ＬＥＤ駆動データ更新処理）

図１１のステップＳ１１４のＬＥＤ駆動データ更新処理を説明する。
この処理は、ランプデータ登録情報に登録されている点灯ナンバ（登録点灯ナンバ（lmpNew）、実行点灯ナンバ（lmpNo））に対応するランプデータテーブルを参照して、ＬＥＤ駆動データを作成する処理である。なお上述のように、ランプデータ登録情報の点灯ナンバには、元々はランプサブシナリオテーブルに記述された、点灯パターンを示すランプナンバがセットされる。ランプナンバは点灯パターンを示すナンバと述べたが、具体的には図２８Ａで述べるランプデータテーブルのランプデータの番号を示すことになる。
【０１９７】
図２４はＬＥＤ駆動データ更新処理を示している。
演出制御ＣＰＵ２００はステップＳ７０１でそれまで出力データとしていたＬＥＤ駆動データをクリアする。
そしてループ処理ＬＰ２として、ランプデータ登録情報のランプチャネルｄｗＣＨ０～ｄｗＣＨ１５のそれぞれについて、ステップＳ７０２～Ｓ７２０の処理が行われる。以下、処理対象のランプチャネルを「ｄｗＣＨｎ」と表記して説明する。
【０１９８】
ステップＳ７０２では演出制御ＣＰＵ２００は対象のランプチャネルｄｗＣＨｎにおける実行点灯ナンバ（lmpNo）と登録点灯ナンバ（lmpNew）が一致しているか否かを確認する。先の図１９ＢのステップＳ６５３のように点灯パターンナンバの登録が行われるため、最初は一致していない。一致していなければ点灯開始として、ステップＳ７０３で、実行点灯ナンバ（lmpNo）に登録点灯ナンバ（lmpNew）の値を代入する。またステップＳ７０４で、実行ライン（ofset）を０にセットし、また実行時間（time）を０にセットする。

なお実行点灯ナンバ（lmpNo）と登録点灯ナンバ（lmpNew）が一致していれば、既に過去に以上のステップＳ７０３，Ｓ７０４の処理が行われたものであるため、これらの処理は不要である。

【０１９９】

或るランプチャネルｄｗＣＨに登録された情報については、登録後、このＬＥＤ駆動データ更新処理の機会毎に、その情報が反映されて各ＬＥＤドライバ９０に出力するＬＥＤ駆動データが作成されていく。

図２６Ａは、図２０Ｂに示したワークのランプデータ登録情報において、ランプチャネルｄｗＣＨ０、ｄｗＣＨ５、ｄｗＣＨ８に登録が行われている状態を示している。この図の状態は、図２６Ｂの時点ｔ０の状態の一例である。

即ちランプチャネルｄｗＣＨ０、ｄｗＣＨ５の情報については、時点ｔ０より以前に、ＬＥＤ駆動データに反映されている。ランプチャネルｄｗＣＨ０については、既に実行点灯ナンバ（lmpNo）と登録点灯ナンバ（lmpNew）が一致され、また実行ライン（ofset）は３（３ライン目）まで進んでいる。またランプチャネルｄｗＣＨ５についても、既に実行点灯ナンバ（lmpNo）と登録点灯ナンバ（lmpNew）が一致され、また実行ライン（ofset）は２（２ライン目）まで進んでいる。時点ｔ０では、ランプチャネルｄｗＣＨ８は登録直後であり、まだ実行点灯ナンバ（lmpNo）と登録点灯ナンバ（lmpNew）が一致されていない。この後、上記ステップＳ７０３，Ｓ７０４の処理が行われることになる。

【０２００】

なお前述したように本実施形態では、ランプチャネルｄｗＣＨ０〜ｄｗＣＨ１５には優先順位が設定されており、ランプチャネルｄｗＣＨ０からｄｗＣＨ１５に向かって順にプライオリティが高くなる。図２６Ａに示すように、優先度の高いランプチャネルｄｗＣＨ１５はエラー報知用に使用される。ランプチャネルｄｗＣＨ１２，ｄｗＣＨ１３，ｄｗＣＨ１４等、優先度が高いランプチャネルは、連続予告や確定予告等、比較的信頼度の高い演出用などに用いられる。また優先度が中程度のランプチャネルは可動体演出用、優先度が比較的低いランプチャネルは、会話予告やステップアップ予告など比較的信頼度の低い演出用に用いられ、さらに優先度が低いランプチャネルは、通常変動、リーチ等に同期したランプ演出に用いられる。

このような優先度の設定のため、図２６Ｂのように複数のランプチャネルｄｗＣＨの動作が重なる場合、優先度の高いランプチャネルの点灯動作が実行される。例えば時点ｔ０からは、ランプチャネルｄｗＣＨ０、ｄｗＣＨ５に基づく点灯動作が制限され、ｄｗＣＨ８に基づく点灯が行われるようにＬＥＤ駆動データが生成される。優先度の低いランプチャネルの情報を反映させないようにするためには、後述するマスクデータが使用される。

【０２０１】

続いて図２４のステップＳ７０５では、演出制御ＣＰＵ２００は実行点灯ナンバ(lmpNo)の値が正常範囲であるか否かを確認する。正常範囲とは、図２８Ａに示すランプデータテーブルのランプデータとして番号が存在する範囲である。

実行点灯ナンバ（lmpNo）が異常であれば、ステップＳ７１９に進む。なおランプチャネルｄｗＣＨｎが登録されていない空きチャネルの場合も、ここでは異常としてステップＳ７１９に進む。

【０２０２】

実行点灯ナンバ（lmpNo）が正常であれば、ステップＳ７０６に進む。ステップＳ７０６では演出制御ＣＰＵ２００は、ランプチャネルｄｗＣＨｎに登録された実行点灯ナンバ（lmpNo）、実行ライン（ofset）に対応する、ランプデータテーブルのアドレスを特定する。

またステップＳ７０７では、ランプチャネルｄｗＣＨｎに登録された実行点灯ナンバ（lmpNo）、マスクデータテーブルのアドレスを特定する。

アドレスの特定のためには演出制御ＣＰＵ２００は、図２７のようなランプデータアドレステーブルを参照する。このランプデータアドレステーブルには、各点灯パターン、例えば全体点滅、右側点滅、左側点滅、役物点灯などを実現するためのランプデータ番号の

アドレスが示されている。図２７の左端の数字は、図２３Ａのランプサブシナリオテーブルで示されるランプナンバであり、例えばランプナンバ２のランプデータのアドレスの欄には、全体点滅の点灯パターンを行うためのランプデータ番号が記憶されたアドレスが記述されている。

また、マスクデータのアドレスの欄には、そのランプデータ番号の点灯パターンを実行する際に必要なマスクデータが記憶されたアドレスが記憶されている。例えば点灯ナンバ５の右側点滅の点灯パターンを行う際には、センターケースマスクが必要になるが、そのセンターケースマスクを行うためのマスクデータのアドレスが記述されている。

【０２０３】

－先行例におけるランプデータテーブルの構造－

図２８Ａにはランプデータテーブルの一部としてランプデータ１，２を示している。

各番号のランプデータにおける各ラインには、タイマ（frame）としての時間データと点灯データが記述されている。タイマ（frame）は各ラインの点灯データによるＬＥＤ駆動データの生成を行う時間を規定する。

タイマ（frame）の数値は、１割込処理分の時間（１６ｍｓ）を単位とした数値である。つまり、図の例において、タイマ（frame）＝「４」のラインは、１６ｍｓ×４＝６４ｍｓに相当する時間を表すものである。

以下、ランプデータテーブルにおいて、各番号のランプデータが書き込まれた各ラインのことを「ラインＬ」と表記する。

【０２０４】

点灯データは、各ＬＥＤドライバ９０に対応して記述されている。先に図４では、枠ドライバ部６１にはｎ個、盤ドライバ部６２にはｍ個のＬＥＤドライバ９０が存在するとしたが、その場合、ランプデータの各ラインＬには、図２８Ａに示すように、（ｎ＋ｍ）個のＬＥＤドライバ９０のそれぞれに対応して点灯データが記述される。なお図では各ＬＥＤドライバ９０との対応をＬＥＤドライバ９０のスレーブアドレスでｗ１～ｗ（ｎ）、ｂ１～ｂ（ｍ）により示している。

点灯データとしては、図４，図５で説明した、ＬＥＤドライバ９０の１つの電流端子９６（１系列のＬＥＤ駆動電流生成）に対して４ビット（０ｈ～Ｆｈ）が割り当てられ（「ｈ」は１６進表記を示す）、１６階調の輝度を指定するようにされている。図５で述べたように、ＬＥＤドライバ９０には２４個の電流端子９６－１～９６－２４がある。このため、１つの点灯データは、「ＦＦＦＦ００００５５５５００００ＡＡＡＡＡＡＡＡ（ｈ）」のように、（４×２４）ビットの情報となる。仮に図示のランプデータ１の１ライン目のように、ＬＥＤドライバ９０（ｗ１）についての点灯データが、「ＦＦＦＦ００００００００００００５５５５００００（ｈ）」であれば、ＬＥＤドライバ９０（ｗ１）の電流端子９６－１～９６－４からは最大輝度「Ｆ」を発光させるための駆動電流を出力し、電流端子９６－５～９６－１６及び９６－２１～９６－２４は非発光（最低輝度）、電流端子９６－１７～９６－２０は輝度「５」としての駆動電流を出力することを指定する情報となる。

ランプデータにはこのような点灯データが、各ＬＥＤドライバ９０（ｗ１～ｗ（ｎ）、ｂ１～ｂ（ｍ））のそれぞれに対して設定されて、かつそれらがラインＬ毎に設定されることで、時系列的に変化する所定の発光パターンが示されることになる。（図では点灯データは一部のみに例示した。空欄部分は図示を省略したもので、実際には点灯データが記述される）

【０２０５】

図２８Ｂにはマスクデータテーブルの例として、マスクデータ１～５を示している。各マスクデータは、枠側の発光部２０ｗの駆動についてのマスクのため、盤側の発光部２０ｂの駆動についてのマスクのため、全体のマスクのため、センターケースのマスクのため、役物のマスクのためなどとして、必要なマスクパターンが記憶されている。

各マスクデータ１～５は、それぞれ各ＬＥＤドライバ９０（ｗ１～ｗ（ｎ）、ｂ１～ｂ

（ｍ））の各電流端子９６－１～９６－２４について、消灯を「０ｈ」、マスク無しを「Ｆｈ」で示すデータとされている。

例えばマスクデータ１についてみると、枠ドライバ部６１におけるＬＥＤドライバ９０（ｗ１～ｗ（ｎ））については、「００００・・・・・００（ｈ）」と、各電流端子９６－１～９６－２４について消灯が設定され、盤ドライバ部６２のＬＥＤドライバ９０（ｂ１～ｂ（ｍ））については、「ＦＦＦＦ・・・・・ＦＦ（ｈ）」と、各電流端子９６－１～９６－２４についてマスク無しが設定されている。つまり枠側の発光部２０ｗのみをマスクすることを指定するデータとなる。（なお、マスクデータ４，５はデータ値の図示を省略している）

【０２０６】

－ＬＥＤ駆動データ更新処理の続き－

図２４のステップＳ７０６，Ｓ７０７では、このようなランプデータテーブル、マスクデータテーブルにおいて、処理中のランプチャネルｄｗＣＨｎに対応するアドレスを特定するものである。

【０２０７】

続いてステップＳ７０８で演出制御ＣＰＵ２００は、ランプデータテーブルの該当ライン（現在対象のランプチャネルｄｗＣＨｎの実行ライン（offset）で示されるラインＬ）のタイマ（frame）を取得し、これを変数Dtimeに代入する。

【０２０８】

ステップＳ７０９で演出制御ＣＰＵ２００は、上記ステップＳ７０６，Ｓ７０７で特定した該当の点灯データとマスクデータを展開し、ＬＥＤ駆動データを生成していく。この処理については後述するが、現在処理中のランプチャネルｄｗＣＨｎについての点灯データとマスクデータを、出力するＬＥＤ駆動データに反映させる処理となる。

【０２０９】

ステップＳ７１０ではランプチャネルｄｗＣＨｎの情報である実行時間（time）を＋１する。

ステップＳ７１１では変数Dtimeと実行時間（time）を比較する。変数Dtimeには現在実行中のラインのタイマ（frame）が代入されている。実行時間（time）はＬＥＤ駆動データ更新処理毎（１６ｍｓ毎）に上記ステップＳ７１０で＋１される。従ってDtime≦timeとなれば、現ラインの終了タイミングとなる。Dtime≦timeではなければ、まだ現在のラインの終了に至らないとしてステップＳ７１１からＳ７１９に進む。Dtime≦timeの場合は、現在のラインの終了としてステップＳ７１１からＳ７１２に進む。

演出制御ＣＰＵ２００はステップＳ７１２で実行時間（time）を０にリセットする。またステップＳ７１３で、実行ライン（offset）の値を＋１する。つまり次のラインが対象となるようにする。

ステップＳ７１４では演出制御ＣＰＵ２００は、実行ライン（offset）に対応するランプデータテーブルの該当ラインのアドレスを特定する。そしてステップＳ７１５，Ｓ７１７で、そのラインに終了コード（D_DTEND）が記述されているか、ループコード（LMP_LP）が記述されているかを確認する。

終了コード（D_DTEND）が記述されていた場合は、ステップＳ７１６で、当該ランプチャネルｄｗＣＨｎの登録点灯ナンバを０に更新する。つまりワーク上で、当該ランプチャネルｄｗＣＨの登録に応じた処理が完了したことを示すようにする。

ループコード（LMP_LP）が記述されていた場合はステップＳ７１８で、実行ライン（offset）の値をループ先のアドレスに更新する。

【０２１０】

ステップＳ７１９で演出制御ＣＰＵ２００は、ランプチャネルｄｗＣＨｎのＬＥＤ駆動データのチェックサムを算出する。またステップＳ７２０でバックアップ用データを保存する。

以上で、１つのランプチャネルｄｗＣＨｎを対象としたＬＥＤデータ更新を終える。ル

ープ処理ＬＰ２として、以上の処理をランプチャネルｄｗＣＨ０～ｄｗＣＨ１５まで順次行うことになる。

【０２１１】

各ランプチャネルｄｗＣＨｎでのＬＥＤ駆動データの更新は、ステップＳ７０９で行われるが、これは次のような処理となる。

図２９Ａは点灯データとマスクデータを展開して生成したＬＥＤ駆動データを記憶する出力データバッファ（発光駆動データ記憶部）を模式的に示している。

出力データバッファは、ランプデータテーブル及びマスクデータテーブルと同様に、各ＬＥＤドライバ９０（ｗ１～ｗ（ｎ）、ｂ１～ｂ（ｍ））のそれぞれに対応して用意される。図示する１つのマスは、１つのＬＥＤドライバ９０に対応する（４×２４）ビットの情報のバッファ領域を示している。

【０２１２】

このような出力データバッファに対して、ステップＳ７０９では、マスクデータをアンド（論理積）で展開し、点灯データをオア（論理和）で展開する処理が行われる。これがループ処理（ＬＰ２）により、各ランプチャネルｄｗＣＨ０～ｄｗＣＨ１５について順次ステップＳ７０９で行われる。

例えば図２６で示した時点ｔ０の場合、出力データバッファは、ランプデータ２のマスクデータ（図２７参照：マスクなし）がアンド展開され、ランプデータ２の３ライン目の点灯データがオア展開された状態の後、ランプデータ５のマスクデータ（図２７参照：センターケースマスク）がアンド展開され、ランプデータ５の２ライン目の点灯データがオア展開された状態となっている。

上述の通りマスクデータは０ｈ（＝００００）が消灯、Ｆｈ（＝１１１１）がマスクなしである。従ってマスクデータをアンドで展開するということは、その時点までの出力データバッファの値について、消灯（マスク）したいビットを「０」とし、マスクしないビットは変更しないということになる。

さらに点灯データをオアで展開するということは、処理中のランプチャネルｄｗＣＨの点灯データが、マスク後の出力データバッファに反映されるということになる。

この処理をランプチャネルｄｗＣＨ０～ｄｗＣＨ１５について順次行うことで、高い番号のランプチャネルほど優先されたＬＥＤ駆動データが出力データバッファ上に形成されることになる。

【０２１３】

このようにランプデータ登録情報（図２０Ｂ参照）には、複数の発光演出に対応する複数種類のランプデータが、優先順位が設定された複数の階層にセットされる場合に、優先順位が低い順にランプデータを反映させていって、発光演出用のＬＥＤ駆動データを作成することで、簡易な処理で優先順位を反映した発光演出用のＬＥＤ駆動データ生成が可能となる。

【０２１４】

（1-6-7：ＬＥＤ出力処理）

上述のようにＬＥＤデータ更新処理で生成されたＬＥＤ駆動データは、図１１のステップＳ１１５で、実際に各ＬＥＤドライバ９０に出力される。

この場合の演出制御ＣＰＵ２００は、ＬＥＤ駆動データの一送信単位のシリアルデータを、第１系統、第２系統の駆動信号出力手段（枠ドライバ部６１と盤ドライバ部６２）に対して略同時的に出力する。

例えば図２９Ｂに示すように、シリアルデータ送信チャネルｃｈ１により枠ドライバ部６１のＬＥＤドライバ９０（ｗ１～ｗ（ｎ））にＬＥＤ駆動データの送信を行い、シリアルデータ送信チャネルｃｈ２により盤ドライバ部６２のＬＥＤドライバ９０（ｂ１～ｂ（ｍ））にＬＥＤ駆動データの送信を行う。

なお、シリアルデータ送信チャネルｃｈ１、ｃｈ２によるＬＥＤ駆動データの送信を並行して同時的に行うのはあくまで一例であり、必ずしも同時的でなくても良い。

【０２１５】
図３０は、１つのＬＥＤドライバ９０に送信するＬＥＤ駆動データのシリアルデータ構造を示している。
例えば図２９Ｂに示した、各ＬＥＤドライバ９０（ｗ１～ｗ（ｎ）、ｂ１～ｂ（ｍ））のそれぞれに対しては、この図３０の構造のシリアルデータでＬＥＤ駆動データが送信される。なおこの図３０は、図４、図５で説明したシリアルデータDATA及びイネーブル信号ENABLEを示している。
このシリアルデータは、イネーブル信号ENABLEがＯＮとされた期間に、１バイト毎順次送信が行われる。つまり当該構造のシリアルデータは、各１バイトが一送信単位となる。
【０２１６】
当該シリアルデータDATAは、１バイトのスレーブアドレス、１バイトのデータ設定開始レジスタアドレス（ＳＸ）、各１バイトの、アドレス（ＳＸ）への書込データ、アドレス（ＳＸ＋１）への書き込みデータ、・・・アドレス（ＳＸ＋２３）への書き込みデータの合計２６バイトで構成される。
このようなシリアルデータDATAは、イネーブル信号ENABLEで規定される期間に、ＬＥＤドライバ９０に取り込まれる。
【０２１７】
「スレーブアドレス」は各ＬＥＤドライバ９０のデバイスＩＤであり、ｗ１～ｗ（ｎ）、又はｂ１～ｂ（ｍ）の別を識別する１バイトコードである。
「データ設定開始レジスタアドレス（ＳＸ）」は、ＬＥＤドライバ９０内のデータ取り込みのためのレジスタの先頭を指定する情報である。例えば図５のデータバッファ／ＰＷＭコントローラ９３の内部レジスタの指定情報である。本実施形態で採用するＬＥＤドライバ９０は、シリアルデータ取り込みのために例えばレジスタアドレス００ｈ～２Ｃｈのレジスタ（各８ビット）が用意される。そのうちのレジスタアドレス１５ｈ～２Ｃｈの２４個のレジスタ（各８ビット）は、２４個の電流端子９６－１～９６－２４に対応した点灯データ（ＰＷＭ階調指示データ）を入力するレジスタ（階調値設定部）とされている。そのため「データ設定開始レジスタアドレス（ＳＸ）」としてはレジスタアドレス１５ｈを指定する。
「アドレス（ＳＸ）への書込データ」としては、「データ設定開始レジスタアドレス（ＳＸ）」で指定されたレジスタアドレス（本実施形態では「１５ｈ」）のレジスタへの書き込みデータ、即ち電流端子９６－１に対応するＬＥＤ駆動データがセットされる。
「アドレス（ＳＸ＋１）への書込データ」としては、「データ設定開始レジスタアドレス（ＳＸ）」で指定されたレジスタアドレス＋１（本実施形態では「１６ｈ」）のレジスタへの書き込みデータ、即ち電流端子９６－２に対応するＬＥＤ駆動データがセットされる。
以下同様であり、「アドレス（ＳＸ＋２３）への書込データ」としては、「データ設定開始レジスタアドレス（ＳＸ）」で指定されたレジスタアドレス＋２３（本実施形態では「２Ｃｈ」）のレジスタへの書き込みデータ、即ち電流端子９６－２４に対応するＬＥＤ駆動データがセットされる。
【０２１８】
なお、これら書き込みデータは８ビット構造であるが、上述のように１つの点灯データは「０ｈ（＝００００）」～「Ｆｈ（＝１１１１）」の４ビット構造である。４ビットで１６階調の輝度を表す。一方、ＰＷＭ階調指示データを入力するレジスタアドレス１５ｈ～２Ｃｈの各レジスタ（各８ビット）は、８ビットで２５６階調に対応可能である。
本実施形態の場合、点灯データは１６階調のデータとするため、レジスタアドレス１５ｈ～２Ｃｈの各レジスタへの８ビットの書き込みデータとしては、上位４ビットに、点灯データをセットするようにしている。この点については後述する。
また、図１１のステップＳ１１３ではＬＥＤドライバ９０の初期化を行うと述べたが、これは具体的には、使用しないレジスタアドレス００ｈ～１４ｈのレジスタにデフォルト値をセットする送信処理となる。この点についても後述する。

【０２１９】
図２５で、１６ｍｓ処理のステップＳ１１５で行われるＬＥＤ駆動データ出力処理を説明する。
演出制御ＣＰＵ２００は図２５のステップＳ８０１で、スタート値に対応するシリアルデータ出力チャネルｃｈ１、ｃｈ２のＬＥＤ駆動データ（図２９Ａの出力データバッファ）の先頭アドレスを特定する。
スタート値とは、最初に送信するＬＥＤドライバ９０の値である。本実施形態ではシリアルデータ出力チャネルｃｈ１、ｃｈ２からそれぞれ、ＬＥＤドライバ９０（ｗ１）、ＬＥＤドライバ９０（ｂ１）に対して最初に送信を行うため、ここでは図２９Ａの出力データバッファにおいてｗ１用のＬＥＤ駆動データが格納された先頭アドレスと、ｂ１用のＬＥＤ駆動データが格納された先頭アドレスを特定することになる。
【０２２０】
次にステップＳ８０２で演出制御ＣＰＵ２００は、スタート値に対応する、シリアルデータ出力チャネルｃｈ１、ｃｈ２のスレーブアドレスを特定する。即ち「ｗ１」「ｂ１」を示すアドレスコードである。
【０２２１】
そしてループ処理ＬＰ３として、ドライバナンバ個数分、ステップＳ８０３～Ｓ８１５の処理をループして実行する。
シリアルデータ出力チャネルｃｈ１からは枠ドライバ部６１のｎ個のＬＥＤドライバ９０に送信を行い、シリアルデータ出力チャネルｃｈ２からは盤ドライバ部６２のｍ個のＬＥＤドライバ９０に送信を行うものであるため、ループ処理の回数はｎ回又はｍ回（ｎ、ｍの多い方の値）となる。仮にｎ＜ｍとすると、ｍ回となる。
【０２２２】
ステップＳ８０３では、演出制御ＣＰＵ２００はシリアルデータ出力チャネルｃｈ１，ｃｈ２におけるイネーブル信号ENABLEをＯＮとする。
そしてステップＳ８０４で演出制御ＣＰＵ２００は、シリアルデータ出力チャネルｃｈ１，ｃｈ２のシリアルデータDATAの出力として、最初の一送信単位（１バイト）のデータ送信、即ちこの場合は図３０のシリアルデータ構造に示した先頭１バイトのスレーブアドレスを送信出力する。シリアルデータ出力チャネルｃｈ１，ｃｈ２については並行して同時的にデータ送信を行う。
その一送信単位の送信を行ったら、ステップＳ８０５，Ｓ８０６で、各シリアルデータ出力チャネルｃｈ１，ｃｈ２についての送信出力完了を待機する。
【０２２３】
シリアルデータ出力チャネルｃｈ１，ｃｈ２についての送信出力完了が確認されたら、演出制御ＣＰＵ２００はステップＳ８０７で、次の一送信単位（１バイト）のデータ送信として、データ設定開始レジスタアドレス（ＳＸ）（本例の場合１５ｈ）を送信出力する。この場合も、シリアルデータ出力チャネルｃｈ１，ｃｈ２について並行して同時的にデータ送信を行う。
その一送信単位の送信を行ったら、ステップＳ８０８，Ｓ８０９で、各シリアルデータ出力チャネルｃｈ１，ｃｈ２についての送信出力完了を待機する。
【０２２４】
ステップＳ８０８，Ｓ８０９でシリアルデータ出力チャネルｃｈ１，ｃｈ２についての送信出力完了が確認されたら、演出制御ＣＰＵ２００はステップＳ８１０で、一送信単位（１バイト）のデータ送信として、レジスタアドレス（ＳＸ＝１５ｈ）～（ＳＸ＋２３＝２Ｃｈ）への書込データ（＝ＬＥＤ駆動データ）を送信出力する。
まずはレジスタアドレス（１５ｈ）への１バイトのＬＥＤ駆動データ送信を、シリアルデータ出力チャネルｃｈ１，ｃｈ２について並行して同時的に行う。そしてステップＳ８１１，Ｓ８１２で、各シリアルデータ出力チャネルｃｈ１，ｃｈ２についての送信出力完了を待機する。
送信完了を確認したら、ステップＳ８１３で、合計２４バイト、つまりレジスタアドレ

ス１５ｈ～２ＣｈまでのＬＥＤ駆動データ送信完了が確認されるまで、１バイトずつ、ステップＳ８１０の送信を実行していく。従って、次はレジスタアドレス（１６ｈ）への１バイトのＬＥＤ駆動データ送信を、シリアルデータ出力チャネルｃｈ１，ｃｈ２について行う。さらにその送信確認後、レジスタアドレス（１７ｈ）へのＬＥＤ駆動データ送信が行われる。以降も、送信完了待機→レジスタアドレス（１８ｈ）へのＬＥＤ駆動データ送信→送信完了待機→レジスタアドレス（１９ｈ）へのＬＥＤ駆動データ送信→送信完了待機・・・→レジスタアドレス（２Ｃｈ）へのＬＥＤ駆動データ送信→送信完了待機という処理が行われていく。

【０２２５】

ＬＥＤ駆動データについて２４バイトの送信出力が完了したら、図３０の構造の１つのＬＥＤドライバ９０に対するシリアルデータ出力が完了したことになる。そこでステップＳ８１４でイネーブル信号ENABLEをオフとし、次に対象とするスレーブアドレスを＋１する。そしてループ処理の１回を終える。

図３１は、以上の１回のループ処理で送信されるデータを示している。シリアルデータ出力チャネルｃｈ１，ｃｈ２において図示のように並列的にスレーブアドレス、データ設定開始レジスタアドレス、ＬＥＤ駆動データＬＰＤＴ１，ＬＰＤＴ２・・・ＬＰＤＴ２４が順次送信される。

このような送信が行われ、ＬＥＤドライバ９０にＬＥＤ駆動データがセットされる。ＬＥＤドライバ９０側では、各レジスタ（１５ｈ～２Ｃｈ）に書き込まれたＬＥＤ駆動データＬＰＤＴ１，ＬＰＤＴ２・・・ＬＰＤＴ２４に示される発光輝度（０ｈ～Ｆｈの１６階調）に基づいて、各電流端子９６－１～９６－２４の電流量が制御され、各電流端子９６－１～９６－２４に接続された各系列のＬＥＤ１２０の発光が行われる。

【０２２６】

続いて２回目のループ処理として、ＬＥＤドライバ９０（ｗ２）、ＬＥＤドライバ９０（ｂ２）について、同様にステップＳ８０３～Ｓ８１５の処理が行われる。さらにその後、３回目のループ処理として、ＬＥＤドライバ９０（ｗ３）、ＬＥＤドライバ９０（ｂ３）について、同様にステップＳ８０３～Ｓ８１５の処理が行われる。

以降、ＬＥＤドライバ９０（ｗ（ｎ））、ＬＥＤドライバ９０（ｂ（ｍ））の全てに送信するまで、ループ処理ＬＰ３が繰り返される。なお、ｎ＜ｍの場合、枠ドライバ部６１のｎ個のＬＥＤドライバ９０の全てにシリアルデータ送信を完了した後は、盤ドライバ部６２の残りのＬＥＤドライバ９０に対してのみステップＳ８０３～Ｓ８１５の処理が行われることになる。

ｍ回のループ処理ＬＰ３で図２９Ｂに示したような、（ｎ＋ｍ）個の各ＬＥＤドライバ９０へのシリアルデータ出力が実行され、ＬＥＤ駆動データ出力（図１１のステップＳ１１５の処理）が完了する。なお、搭載されるＬＥＤドライバ９０の数が異なれば、当然ループ回数も変化する。

また以上の例は２つのシリアルデータ出力チャネルｃｈ１，ｃｈ２を用いる例で述べているが、３以上のシリアルデータ出力チャネルを利用する場合も、それらが並行して同時的にシリアルデータ送信が行われるようにすれば良い。

【０２２７】

以上の図２５の処理のように、演出制御ＣＰＵ２００は、第１系統（シリアルデータ出力チャネルｃｈ１）と第２系統（シリアルデータ出力チャネルｃｈ２）の駆動信号出力手段（枠ドライバ部６１と盤ドライバ部６２）には、一送信単位のシリアルデータを同時的に送信する。そして当該送信の完了後、次の一送信単位のシリアルデータを、第１系統と第２系統の駆動信号出力手段に対して略同時的に送信する。このような処理で、ＬＥＤ駆動データを各ＬＥＤドライバ９０へ送信していく。

これにより全体のシリアルデータ送信動作に要する時間を短縮でき、もって演出制御ＣＰＵ２００の処理負担を軽減できる。

【０２２８】

この理由を説明する。

図３２は演出制御ＣＰＵ２００がシリアルデータ送信に使用するクロック信号CLK、送信データ、ＴＤＲＥ（送信データエンプティフラグ）、及び送信データの送信データレジスタへの書込タイミングを示している。

送信データとは、上述のスレーブアドレス、データ設定開始レジスタアドレス、ＬＥＤ駆動データＬＰＤＴ１，ＬＰＤＴ２・・・ＬＰＤＴ２４に相当する。ここでは１送信単位としての８ビットをデータｄ０～ｄ７として示している。

【０２２９】

演出制御ＣＰＵ２００は、図２５のステップＳ８０４，Ｓ８０７，Ｓ８１０のそれぞれで一送信単位の８ビットデータを送信する場合、各シリアルデータ出力チャネルｃｈ１，ｃｈ２における送信データレジスタに８ビットの送信データを書き込む。送信データレジスタへ書き込まれた送信データは、クロック信号CLKのタイミング毎に１ビットずつ送信出力される。

ここで、ＴＤＲＥは、送信データレジスタに送信データを書き込むと「０」となり、送信データレジスタに有効なデータが存在していることを示す。そして送信データレジスタのデータが実際に送信出力（送信用シフトレジスタへのロード）されて送信が開始されると「１」になり、送信データレジスタに有効なデータが存在しないことを示す。

従って演出制御ＣＰＵ２００の送信処理としては、ＴＤＲＥが「１」となったことを確認して、８ビットの送信データを送信データレジスタへ書き込み、その次の８ビットの送信データの書き込みは、次にＴＤＲＥが「１」となるまで待機する。即ち図３２に待機時間ＴＷとして示す期間は、次の送信データの処理ができないことになる。

【０２３０】

シリアルデータ送信に関し、従来は、１つのシリアルデータ出力チャネルから例えば第１系統の駆動信号出力手段に一送信単位（８ビット）のシリアルデータを送信し、当該送信の完了後、次の送信チャネルから第２系統の駆動信号出力手段に一送信単位（８ビット）のシリアルデータを送信していく、というように送信動作を行っていた。例えば図２５のステップＳ８０４～Ｓ８０６に相当する処理は、従来通常は、

（ＳＴ１）ｃｈ１からスレーブアドレス出力（送信データレジスタへの書込）
（ＳＴ２）送信出力完了待機
（ＳＴ３）ｃｈ１からデータ設定開始レジスタアドレス出力（送信データレジスタへの書込）
（ＳＴ４）送信出力完了待機
（ＳＴ５）ｃｈ１からＬＥＤ駆動データＬＰＤＴ１出力（送信データレジスタへの書込）
（ＳＴ６）送信出力完了待機
（ＳＴ７）ｃｈ１からＬＥＤ駆動データＬＰＤＴ２出力
（ＳＴ８）送信出力完了待機
・・・中略・・・
（ＳＴ５１）ｃｈ１からＬＥＤ駆動データＬＰＤＴ２４出力
（ＳＴ５２）送信出力完了待機
（ＳＴ５３）ｃｈ２からスレーブアドレス出力
（ＳＴ５４）ｃｈ２送信出力完了待機
（ＳＴ５５）ｃｈ２からデータ設定開始レジスタアドレス出力
（ＳＴ５６）送信出力完了待機
（ＳＴ５７）ｃｈ２からＬＥＤ駆動データＬＰＤＴ１出力
（ＳＴ５８）送信出力完了待機
（ＳＴ５９）ｃｈ２からＬＥＤ駆動データＬＰＤＴ２出力
（ＳＴ６０）送信出力完了待機
・・・中略・・・
（ＳＴ１０３）ｃｈ２からＬＥＤ駆動データＬＰＤＴ２４出力
（ＳＴ１０４）送信出力完了待機

というように行っていた。

この場合、シリアルデータ出力チャネルｃｈ１，ｃｈ２のそれぞれで一送信単位の送信毎に待ち時間が発生し、送信処理効率が悪い。

これに対して本構成では、

（Ｓ８０４）ｃｈ１、ｃｈ２からスレーブアドレス出力（送信データレジスタへの書込）

（Ｓ８０５，Ｓ８０６）ｃｈ１、ｃｈ２送信出力完了待機

というように、同時的に両シリアルデータ出力チャネルｃｈ１，ｃｈ２での送信処理を行うようにすることで、処理が効率化される。

ステップＳ８０７～Ｓ８０９の処理、及びステップＳ８１０～Ｓ８１２の処理についても同様である。

【０２３１】

そして、１つのＬＥＤドライバ９０に対する送信は１ループ処理において１つの送信チャネル毎に２６バイト分（２６回の送信データレジスタへの書込）となる。各シリアルデータ出力チャネルｃｈ１，ｃｈ２について、同時的な処理を実行しなければ、総待機時間は、１回のループ処理期間でほぼ（５２×ＴＷ）となる。これが本実施形態の場合、１回のループ処理期間でのほぼ（２６×ＴＷ）と、著しく短縮できる。

図１１のステップＳ１１５のＬＥＤ駆動データ出力処理において、このようにシリアルデータ出力処理が効率化できることで、演出制御ＣＰＵ２００の処理負担は大きく軽減される。

【０２３２】

また本構成では、シリアルデータ送信されるＬＥＤ駆動データは、発光輝度（階調）を示すデータとされ、上記のように例えば４ビットで１６階調の発光駆動を実現する。このような階調制御を可能とすることで、より多様な演出効果を実現することができる。

なお、発光輝度を１６階調とする例を述べたが、もちろんより多階調としても良い。

【０２３３】

また、パチンコ遊技機１は、枠部（前枠２、外枠４及びガラス扉５）と、この枠部に取り付けられる遊技盤部（遊技盤３）とを備えている。そして第１系統の駆動信号出力手段（枠ドライバ部６１）は、前枠２に設けられた演出手段であるランプ部６３（発光部２０ｗ）にＬＥＤ駆動信号を出力する複数のドライバ回路（ＬＥＤドライバ９０）で構成され、第２系統の駆動信号出力手段（盤ドライバ部６２）は、遊技盤３に設けられた演出手段であるランプ部６４（発光部２０ｂ）にＬＥＤ駆動信号を出力する複数のドライバ回路（ＬＥＤドライバ９０）で構成されている。

このような構成により、枠ドライバ部６１のＬＥＤドライバ９０はすべて枠側の発光部２０ｗに対応し、また盤ドライバ部６２のＬＥＤドライバ９０はすべて盤側の発光部２０ｂに対応することとなり、これにより配線の容易化、効率化、演出制御設定の容易性などを実現できる。

【０２３４】

－点灯データのビット変換について－

ところで、パチンコ遊技機１においては、ステップＳ８１０で、上述のように４ビット構造とされている点灯データを８ビットに変換したものを一送信単位のシリアルデータとしてＬＥＤドライバ９０に出力している。

このビット変換について図３３で説明する。図３３Ａは１つのＬＥＤドライバ９０に送信する４×２４ビットの点灯データの例を示している。即ち図２９Ａの出力データバッファの１マス分に相当するデータの例である。先頭の「Ｆ８ｈ」（図では「ｈ」を省略）は、ＬＥＤドライバ９０のレジスタアドレス１５ｈ、１６ｈに送信すべき点灯データである、つまり「Ｆｈ」はＬＥＤドライバ９０のレジスタアドレス１５ｈに書き込む、電流端子９６－１についての階調値としての点灯データであり、「８ｈ」は、レジスタアドレス１６ｈに書き込む、電流端子９６－２についての階調値としての点灯データである。次の「０Ｆｈ」は、同様にレジスタアドレス１７ｈ、１８ｈに対応する。以下同様で、最後の「００ｈ」は、レジスタアドレス２Ｂｈ、２Ｃｈに対応する。

【０２３５】
ここで図３３Ａの「Ｆ８ｈ」「０Ｆｈ」・・・「００ｈ」のように、演出制御ＣＰＵ２００の内部処理単位としての８ビット単位で区切って考えた場合に、上位４ビット（例えば「Ｆ８ｈ」のうちの「Ｆｈ」）は、奇数レジスタアドレス（例えばレジスタアドレス１５ｈ）に対する点灯データで、下位４ビット（例えば「Ｆ８ｈ」のうちの「８ｈ」）は偶数レジスタアドレス（例えばレジスタアドレス１６ｈ）に対する点灯データとなる。
なお、ここでの「奇数レジスタアドレス」「偶数レジスタアドレス」というのはＬＥＤドライバ９０のレジスタ設計に応じた一例である。あくまで送信先を、レジスタアドレス１５ｈ～２Ｃｈとした場合として便宜的に述べているにすぎない。
この場合に、奇数レジスタアドレス（１５ｈ、１７ｈ・・・２Ｂｈ）への点灯データについては、演出制御ＣＰＵ２００は、８ビットデータの下位４ビットをマスクして、それをＬＥＤドライバ９０に送信する。一方、偶数レジスタアドレス（１６ｈ、１８ｈ・・・２Ｃｈ）への点灯データについては、演出制御ＣＰＵ２００は、８ビットデータの上位４ビットをマスクして、それを４ビットシフトしたものをＬＥＤドライバ９０に送信する。つまり結果的に４ビットの点灯データを８ビットに変換して送信する。
【０２３６】
このビット変換として、「Ｆ８ｈ」の場合を図３３Ｂ、図３３Ｃに示した。
図３３Ｂは、点灯データ「Ｆｈ」をレジスタアドレス１５ｈに送信する場合の処理を示している。図２９Ａの出力データバッファから取り出す点灯データは、各４ビットの２つの点灯データである「Ｆ８ｈ」＝「１１１１１０００」となる。このデータに対し、下位４ビットをマスクする。例えばマスクデータ「Ｆ０ｈ」＝「１１１１００００」を用いてアンドをとる。これにより、図示のように、レジスタアドレス１５ｈに送信するシリアルデータ「１１１１００００」＝「Ｆ０ｈ」が得られることになる。この場合、電流端子９６－１からは最大階調値（Ｆ０ｈ）の駆動電流出力が行われることとなる。
一方、図３３Ｃは、点灯データ「８ｈ」をレジスタアドレス１６ｈに送信する場合の処理を示している。上記のとおり図２９Ａの出力データバッファから取り出した点灯データは「Ｆ８ｈ」＝「１１１１１０００」である。このデータに対し、上位４ビットをマスクする。例えばマスクデータ「０Ｆｈ」＝「００００１１１１」を用いてアンドをとる。これにより、図示のように「００００１０００」というデータが得られる。さらにこの場合、当該データを４ビットシフトすることで、レジスタアドレス１６ｈに送信するシリアルデータ「１０００００００」＝「８０ｈ」が得られることになる。この場合、電流端子９６－１からは中間的な階調値（８０ｈ）の駆動電流出力が行われることとなる。
【０２３７】
以上のように、パチンコ遊技機１では、演出制御部５１は、例えば演出制御ＲＯＭ２０１に第１ビット数（例えば８ビット）より少ないビット数である第２ビット数（例えば４ビット）の発光輝度情報（点灯データ）をランプデータテーブルなどの形式で記憶する。そして演出制御ＣＰＵ２００は、第２ビット数の発光輝度情報を、第１ビット数に変換してＬＥＤ駆動データを生成し、そのＬＥＤ駆動データを発光駆動信号出力手段であるＬＥＤドライバ９０にシリアルデータとして送信する。
ＬＥＤドライバ９０のレジスタアドレス１５ｈ～２Ｃｈが、８ビットのＰＷＭデータを入力可能とされていることは、８ビットの点灯データとすれば２５６階調のＬＥＤ発光制御が可能である。しかしながら、一般に遊技機の発光演出に用いる発光部２０ｗ、２０ｂに対する制御としては、２５６階調は必要ではない。具体的には「００ｈ」「１０ｈ」「２０ｈ」・・・「Ｆ０ｈ」の１６階調で十分である。
そこでパチンコ遊技機１では、点灯データを４ビット構造として、１６階調表現を行うようにする。そしてＬＥＤドライバ９０への出力時には、４ビットを８ビットに変換して送信する。
【０２３８】
このようにすることで、演出制御部５１は内部ＲＯＭ（演出制御ＲＯＭ２０１）に記憶する点灯データのデータ容量を大幅に削減でき、メモリ容量消費を抑制できることになる

。

また、近年の汎用的なＬＥＤドライバは、２５６階調などの高性能タイプのものの方が生産量が多く、価格的にも有利であることが多い。そのため、例えば１６階調が必要な場合に、１６階調の制御能力を持つＬＥＤドライバを採用するよりも、２５６階調のＬＥＤドライバを採用する方が、コストメリットが得られ、又、安定的入手性もよい。これにより製造上及びコスト的なメリットが得られる。

また上述したように、演出制御ＣＰＵ２００は、４ビットの点灯データを８ビットにおける上位４ビットに配置して、８ビットのＬＥＤ駆動データ（送信するシリアルデータ）を生成する。

このように上位４ビット配置により、１６階調のＬＥＤ駆動データを生成でき、またその処理は例えば図３３Ｂ、図３３Ｃの例のように、非常に簡易な処理である。このためビット数変換の処理負担はほとんど問題とならない。

【０２３９】

なお上記例では、８ビットに変換した点灯データを「００ｈ」「１０ｈ」「２０ｈ」・・・「Ｆ０ｈ」とするとしたが、次のような例も考えられる。

この場合、４ビットの点灯データが「Ｆｈ」のときの、８ビットのシリアルデータ「Ｆ０ｈ」が制御上の最大輝度となるが、ＬＥＤドライバ９０は「ＦＦｈ」まで対応できる。そこで、８ビットのシリアルデータの「Ｆ０ｈ」の下位４ビットを「１１１１」に変換して「ＦＦｈ」（＝１１１１１１１１）を送信するようにしてもよい。つまり送信するシリアルデータは「００ｈ」「１０ｈ」「２０ｈ」・・・「Ｅ０ｈ」「ＦＦｈ」とする。このようにすると、ＬＥＤドライバ９０の発光駆動能力上での最大輝度の発光を実行させることが可能となる。

また、「００ｈ」以外の「１０ｈ」「２０ｈ」・・・「Ｆ０ｈ」について、下位４ビットを「１１１１」に変換し、「１Ｆｈ」「２Ｆｈ」・・・「ＦＦｈ」として送信してもよい。これにより、各階調での発光輝度を上げ、演出効果を向上させることができる。

また、ここまでは４ビットの点灯データを８ビットとする例で述べたが、これは一例である。第１ビット数（例えば８ビット）はＬＥＤドライバ側の設計に応じて決められる。また第２ビット数（例えば４ビット）は、表現したい階調に応じて設定すればよい。例えば２階調なら１ビット、４階調なら２ビット、８階調なら３ビット・・・とすれば良い。

また本実施形態では、シリアルデータ出力としてＬＥＤ駆動データ送信を行うものであるが、この第２ビット数の点灯データを第１ビット数に変換して発光駆動ドライバに送信するという処理は、パラレルデータ出力、或いは無線送信出力などにも応用でき、その場合に送信先の階調能力と、求める階調表現の差がある場合に、送信元のデータ記憶容量の低減効果を得ることができる。

【０２４０】

［1-7. 実施形態の発光駆動手法］

（1-7-1. 実施形態の発光駆動情報格納手法）

先行例としてのパチンコ遊技機１においては、シナリオデータ（演出進行データ）に従った発光部２０ｂ、２０ｗの発光動作を実現するために、先の図２８Ａに示したランプデータテーブルのように、ＬＥＤ１２０の点灯データを格納したテーブルを演出制御ＲＯＭ２０１に記憶させている。

上述のように、シナリオデータ（ランプサブシナリオテーブル：図２３Ａ参照）には、ランプデータテーブルにおけるランプデータ番号（ランプナンバ）の情報が含まれ、該ランプデータ番号と実行中のライン（ラインＬ）の情報とに基づき、ランプデータテーブルから各ＬＥＤ１２０の点灯データが読み出されて、それぞれ対応するＬＥＤドライバ９０に出力される。

【０２４１】

ここで、図２８Ａに示したように、先行例におけるランプデータテーブルでは、各ランプデータにおけるラインＬごとに、ＬＥＤ１２０一つ一つの点灯データを格納している。

前述のように、ランプデータテーブルに格納されるＬＥＤ１２０一つあたりの点灯データのデータ容量は４ビットである。また、ＬＥＤドライバ９０一つあたりのＬＥＤ数は２４、ＬＥＤドライバ９０の総数は３２である。

よって、先行例において、ランプデータテーブルにおける一つのラインＬに対しては、

４ビット×２４×３２＝３０７２ビット

のデータが格納される。

さらに、ランプデータテーブルにおけるラインＬの総数をＬｎとすると、ランプデータテーブルの総データ容量（タイマ（frame）のビット数は除く）は、

３０７２Ｌｎビット

と表すことができる。

【０２４２】

先行例におけるランプデータテーブルでは、各ＬＥＤ１２０の点灯データに対して、該点灯データによる発光状態（或いは非発光状態）を継続する時間長の情報、具体的にはタイマ（frame）の情報を対応づけ、これら点灯データとタイマとを対応づけた情報を各ラインＬに格納している。

パチンコ遊技機１において、演出発光動作における発光時間の最小単位は、１割込み処理分の時間（本例では１６ｍｓ）とされている。このため、ランプデータテーブルにおいては、タイマの情報を格納せず、各ラインＬが１割込み処理分の時間を表すものとして扱うことが考えられる。

しかしながら、このように１ラインＬ＝１割込み処理分の時間として各ＬＥＤ１２０の点灯データを格納した場合には、テーブルに格納される点灯データの重複格納量が増大し、データ容量の増加を招く。例えば、ランプナンバで特定された発光パターンが所定のＬＥＤ１２０のみを「６４ｍｓ点灯」→「６４ｍｓ消灯」→「６４ｍｓ点灯」させて点滅させるというパターンであったとする。この場合、「６４ｍｓ点灯」、「６４ｍｓ消灯」、「６４ｍｓ点灯」の各々では、各ＬＥＤ１２０の点灯データは共通となる。具体的に、ラインＬ＝１割込み処理分の時間とした場合、「６４ｍｓ点灯」に対応した四つのラインＬに対してそれぞれ同一の点灯データを格納することになる。同様に「６４ｍｓ消灯」、さらにそれに続く「６４ｍｓ点灯」それぞれについても、対応した四つのラインＬに対してそれぞれ同一の点灯データを格納することになり、テーブルに格納される点灯データの重複格納量が増大してしまう。

【０２４３】

そこで、先行例においては、各ラインＬの点灯データに対してタイマ（frame）の情報、すなわち動作の継続時間長を表す情報を対応づけて格納するようにし、いわば、点灯データを時間方向に圧縮している。これにより、テーブルにおける点灯データの重複格納量の低減が図られ、発光制御に係るメモリ消費量の削減が図られている。

【０２４４】

しかしながら、遊技機においてメモリ（演出制御ＲＯＭ２０１）の記憶容量は有限であり、可能な限り消費量は削減されるべきである。

本実施形態では、発光制御に係るデータ記憶容量の削減を図ることで、遊技機におけるメモリ消費量の削減を図る。

【０２４５】

図３４は、実施形態におけるランプデータテーブルのデータ構造を模式的に表している。

先ず、実施形態のランプデータテーブルにおいても、先行例と同様、各ラインＬにはタイマ（frame）の情報を格納するものとし、時間方向での格納データ圧縮を図っている。

【０２４６】

そして、実施形態のランプデータテーブルでは、各ラインＬに点灯データではなく発光パターン識別子を格納するものとしている（図中「パターン１」「パターン２」等）。

発光パターン識別子は、複数個の発光素子による発光パターンを一意に識別するための識別子である。具体的に、本例では、所定複数個の発光素子の群を一つの単位発光素子群

としたときに、該単位発光素子群の発光パターンを一意に識別するための識別子とされている。本例における単位発光素子群は、一つのＬＥＤドライバ９０により発光駆動されるＬＥＤ群（本例では２４個）として定められている。
【０２４７】
図３４に示すように、本例のランプデータテーブルでは、各ランプナンバのランプデータにおける各ラインＬにおいて、ｗ１～ｗ（ｎ）及びｂ１～ｂ（ｍ）の各ＬＥＤドライバ９０ごとの発光パターン識別子が格納されている。
【０２４８】
ここで、単位発光素子群がとり得る発光パターンは、パチンコ遊技機１で実行され得る発光動作の種類が有限であることから、有限である。発光パターン識別子としては、パチンコ遊技機１で実行される得る発光動作に応じて各単位発光素子群がとり得る発光パターンごとに定義しておく。
以下、このように定義される発光パターン識別子の総数を「Ｘ」と表記する。
【０２４９】
発光演出動作に伴う単位発光素子群ごとの発光パターンは、同一のラインＬ、或いは異なるラインＬ間において重複し得るものである。
例えば、或るランプナンバ（ランプナンバｙとする）に対応する発光パターンが、上記で例示したものと同様、所定のＬＥＤ１２０のみを「６４ｍｓ点灯」「６４ｍｓ消灯」「６４ｍｓ点灯」させるというパターンであったとする。この場合、該ランプナンバｙのランプデータにおいては、１番目のラインＬには所定のＬＥＤ１２０を「６４ｍｓ点灯」させそれ以外のＬＥＤ１２０を消灯させるための単位発光素子群ごとの発光パターン識別子が格納される。２番目のラインＬには、所定のＬＥＤ１２０を「６４ｍｓ消灯」させそれ以外のＬＥＤ１２０を消灯させるための単位発光素子群ごとの発光パターン識別子が格納され、さらに、３番目のラインＬには所定のＬＥＤ１２０を「６４ｍｓ点灯」させそれ以外のＬＥＤ１２０を消灯させるための単位発光素子群ごとの発光パターン識別子が格納される。
このようなケースにおいて、各ラインＬでは、点滅対象のＬＥＤ１２０以外のＬＥＤ１２０については発光パターンが同一とされるため、異なる単位発光素子群の間で発光パターン識別子が重複し得る。また、各ラインＬの間で見ても、点滅対象のＬＥＤ１２０以外のＬＥＤ１２０については発光パターンが同一に維持されるため、該当する単位発光素子群については、ラインＬ間で発光パターン識別子が重複し得る。さらに、１番目と３番目のラインＬ間においては、点滅対象のＬＥＤ１２０の発光パターンは同一であるため、該当する単位発光素子群についてはラインＬ間で発光パターン識別子が重複する。
【０２５０】
このような点からランプデータテーブルにおいては、図３４に例示するように、格納される発光パターン識別子に重複が生じるものである。例えば図中、ランプデータ１（ランプナンバ１に対応したランプデータ）における１番目のラインＬでの「パターン１」の重複や、ランプデータ２における１番目のラインＬと３番目のラインＬの間での「パターン１２０」の重複等である。
なお、発光パターン識別子の重複は、同一ランプナンバのランプデータ中に限らず、異なるランプナンバのランプデータ間においても生じ得るものである（図中「パターンＸ」等を参照）。
【０２５１】
なお、図３４においても、先の図２８Ａと同様にランプデータテーブルに格納される一部のランプナンバに対応したランプデータのみを示したが、ランプデータテーブルにはシナリオデータにより指示され得る全てのランプナンバに対応するランプデータが格納されるものである。
【０２５２】
上記のように実施形態のパチンコ遊技機１では、ランプデータテーブルにおいて、異なる発光素子群について共通の発光パターン識別子（発光パターン情報）が記憶される。

【０２５３】

実施形態のパチンコ遊技機１では、ランプデータテーブルには点灯データそのものではなく発光パターンの種別を識別するための発光パターン情報が格納されているため、該発光パターン情報を点灯データに変換するための変換テーブル（点灯データテーブル）が演出制御ＲＯＭ２０１に格納されている。

【０２５４】

図３５は、変換テーブルのデータ構造を模式的に示している。

変換テーブルにおいては、合計Ｘ個となる発光パターン識別子ごとに、単位発光素子群を構成するＬＥＤ１２０ごとの点灯データが格納されている。

本例においても、先行例と同様、ＬＥＤ１２０一つあたりの点灯データは４ビットデータとされている（つまり先行例と同様に４ビット→８ビット変換を行うことを前提としている）。本例では、単位発光素子群を構成するＬＥＤ数は「２４」であることから、一つの発光パターン識別子に対して格納される点灯データのビット数は「９６」である。

【０２５５】

このような変換テーブルにより、シナリオデータに基づきランプデータテーブルから特定される単位発光素子群ごとの発光パターン識別子を、ＬＥＤ１２０ごとの点灯データに変換することができる。

このようにＬＥＤ１２０ごとの点灯データが得られた以降の処理については先行例の場合と同様である。この点の詳細については後に改めて説明する。

【０２５６】

ここで、仮に、パチンコ遊技機１において単位発光素子群がとり得る発光パターンの種類の数「Ｘ」が「１０００」であるとする。

この場合、発光パターン識別子のビット数としては、１０ビット（１０２４通りの表現が可能）あればよい。

前述のように、ランプデータテーブルにおけるラインＬの総数をＬｎとおくと、この場合におけるランプデータテーブルの総データ容量（タイマ（frame）のビット数は除く）としては、

　１０ビット×３２（ドライバ数）×Ｌｎ＝３２０Ｌｎビット

と表すことができる。

【０２５７】

一方、変換テーブルは、発光パターン識別子（ここでは１０００個）ごとに、単位発光素子群分の点灯データ（データ容量＝９６ビット：４×２４）を格納しておけばよい。従って、変換テーブルの総データ容量は、

　９６ビット×１０００＝９６０００ビット

と表すことができる。

【０２５８】

例えば、ランプデータテーブルにおける総ライン数Ｌｎ＝１０００とすると、先行例におけるランプデータテーブルの総データ容量は、

　３０７２ビット×１０００＝３０７２０００ビット

であり、実施形態におけるランプデータテーブル＋変換テーブルの総データ容量は、

　３２０ビット×１０００＝３２００００ビット＋９６０００ビット＝４１６０００ビット

である。

【０２５９】

このようにＸ＝１０００、Ｌｎ＝１０００の例においては、先行例に対しメモリ消費量を約１３％に抑えることができる。

実施形態において、このようにメモリ消費量が削減されるのは、先行例においては前述したランプデータテーブルにおける発光パターンの重複部分にそれぞれ点灯データを格納していたのに対し（つまり点灯データの重複格納が生じていたのに対し）、実施形態では、該重複部分の点灯データは、該重複部分に格納された発光パターン識別子と対応づけら

れて変換テーブルにおける１箇所にのみ格納されるためである。
　この点から理解されるように、実施形態の発光駆動情報格納手法によれば、点灯データ（発光輝度情報）の重複格納量の抑制が図られるものである。
【０２６０】
　（1-7-2．実施形態のＬＥＤ駆動データ更新処理）

　上記により説明した実施形態としてのランプデータテーブル及び変換テーブルに対応して、この場合の演出制御ＣＰＵ２００は、先の図２４に示したＬＥＤ駆動データ更新処理に代えて、図３６に示すＬＥＤ駆動データ更新処理を実行する。実施形態において、演出制御ＣＰＵ２００が行うＬＥＤ駆動データ更新処理以外の各処理については先の図１１～図２３及び図２５～図３３で説明したものと同様となることから重複説明は避ける。
　なお、図３６において、既に説明済みとなった部分と同様となる部分については同一のステップ番号を付して説明を省略する。
【０２６１】
　図３６に示すように、この場合のＬＥＤ駆動データ更新処理においても、図２４の場合と同様にステップＳ７０１～Ｓ７０８までの処理が行われる。但しこの場合、ランプデータテーブルは図３４で例示したデータ構造とされるため、ステップＳ７０６の特定処理では、特定したアドレスにより、ランプデータテーブルにおける特定のラインＬに格納された単位発光素子群ごとの発光パターン識別子が特定される。
【０２６２】
　この場合の演出制御ＣＰＵ２００は、ステップＳ７０８に続くステップＳ９０１において、発光パターン識別子を点灯データに変換する処理を行う。すなわち、ステップＳ７０６の処理によりアドレスが特定されたラインＬに格納されている単位発光素子群ごとの発光パターン識別子を、図３５に示した変換テーブルに基づきＬＥＤ１２０ごとの点灯データに変換する。
【０２６３】
　この場合の演出制御ＣＰＵ２００は、ステップＳ９０１の変換処理を行ったことに応じてステップＳ７０９に処理を進める。ステップＳ７０９以降の処理については既に説明したものと同様となるため重複説明は避ける。
　このようにランプデータテーブルにおける一つのラインＬ分の点灯データが得られた以降の処理については、先行例の場合と同様となる。
【０２６４】
　実施形態においても、各ＬＥＤ１２０の点灯データは４ビットデータとされるため、演出制御ＣＰＵ２００は、先行例の場合と同様に４ビットの点灯データを８ビットに変換する処理を行う（図３３を参照）。
　このようなビット変換により、各ＬＥＤドライバ９０の各レジスタアドレスに送信すべき８ビットのシリアルデータが得られ、これらシリアルデータがステップＳ８１０の処理（図２５）により対応するレジスタアドレスに出力される。
　これにより、各ＬＥＤ１２０が点灯データに応じた輝度により発光され、或いは非発光とされて、シナリオデータに基づく発光動作が実現される。
【０２６５】
　（1-7-3．発光駆動情報格納手法の変形例）

　なお、上記では単位発光素子群が一つのＬＥＤドライバ９０により発光駆動されるＬＥＤ群とされた例を挙げたが、単位発光素子群の区分はこれに限定されず、多様に考えられる。例えば、単位発光素子群は、所定複数のＬＥＤドライバ９０により発光駆動されるＬＥＤ群として定めることもできる。このとき、メモリ消費量削減を図る上では、単位発光素子群を構成するＬＥＤ１２０の数は、発光部２０ｂ及び２０ｗにおけるＬＥＤ１２０の総数よりも少なくすることが望ましい。
【０２６６】

或いは、発光部２０ｂ、２０ｗそれぞれが、発光色の異なる複数のＬＥＤ１２０で成る発光ユニットを複数有した構成とされた場合には、単位発光素子群は、一つの発光ユニットを構成するＬＥＤ群として区分してもよい。

一例として、発光ユニットとしては、それぞれ発光色がＲ（赤）、Ｇ（緑）、Ｂ（青）とされた三つのＬＥＤ１２０で成るものとし、各ＬＥＤドライバ９０は、２４個の発光ユニットを発光駆動可能に構成されている場合を挙げることができる。

この場合、ＬＥＤドライバ９０の総数が「３２」であれば、発光ユニットの総数は３２×２４より「７６８」であり、ＬＥＤ１２０の総数はその３倍の「２３０４」である。

【０２６７】

図３７は、このような構成を前提とした際に、１発光ユニットを構成するＬＥＤ群を単位発光素子群とした場合のランプデータテーブルのデータ構造を模式的に示している。

なお、先の図３４と同様に、図３７においてもランプデータ１、２に対応した部分のみを例示している。また、図３７では、ランプデータテーブルに格納される発光パターン識別子は一部のみを例示しており、空欄部分は図示を省略したもので、実際には発光パターン識別子が格納されている。

【０２６８】

ここで、上記のように発光部２０ｂと発光部２０ｗとを合わせたＬＥＤ１２０の総数が「２３０４」個とされる場合において、先行例のようにランプデータテーブルにＬＥＤ１２０ごとの点灯データを格納した場合、ランプデータテーブルにおける一つのラインＬに対しては、

４ビット×３×２４×３２＝９２１６ビット

のデータが格納される。

この場合もランプデータテーブルにおけるラインＬの総数をＬｎとすると、ランプデータテーブルの総データ容量（タイマ（frame）のビット数は除く）は、

９２１６Ｌｎビット

と表される。

【０２６９】

一方、図３７に示した格納手法を採った場合、ランプデータテーブルの総データ容量（タイマ（frame）のビット数は除く）は、単位発光素子群がとり得る発光パターンの種類の数「Ｘ」が例えば「１０００」であり、これに対応した発光パターン識別子のビット数が１０ビットであるとすると、

１０ビット×２４（１ドライバの発光ユニット数）×３２（ドライバ数）×Ｌｎ＝７６８０Ｌｎビット

と表される。

【０２７０】

また、この場合も変換テーブルは、発光パターン識別子（ここでは１０００個）ごとに、単位発光素子群分の点灯データ（データ容量＝１２ビット：４×３）を格納しておけばよく、総データ容量は、

１２ビット×１０００＝１２０００ビット

と表される。

【０２７１】

この場合も、ランプデータテーブルにおける総ライン数Ｌｎが例えば「１０００」であるとすると、先行例の格納手法によるランプデータテーブルの総データ容量は、

９２１６ビット×１０００＝９２１６０００ビット

であり、図３７に示した格納手法によるランプデータテーブル＋変換テーブルの総データ容量は、

７６８０ビット×１０００＝７６８００００ビット＋１２０００ビット＝７６９２０００ビット

である。

【０２７２】

このようにＸ＝１０００、Ｌｎ＝１０００の例においては、先行例の格納手法に対しメモリ消費量を約８３％に抑えることができる。
【０２７３】
なお、先の図３４に示したデータ格納手法も含め、「Ｘ」の数値はあくまで説明上の例を示したものに過ぎない。「Ｘ」の数値は、パチンコ遊技機１が行い得る発光動作の種類や、異種の発光動作間における部分的な発光パターンの一致数等に応じて上下するものである。具体的には、上記発光動作の種類の数が少ないほど「Ｘ」の数値が減少する傾向となる。また、上記異種の発光動作間における部分的な発光パターンの一致数が多いほど「Ｘ」の数値が減少する傾向となる。「Ｘ」の数値が少なくなるほど、メモリ消費量の削減効果は増大する。
【０２７４】
［1-8. 第一実施形態のまとめ及び変形例］

以上で説明した実施形態の遊技機（パチンコ遊技機１）によれば、発光制御のための処理を効率化でき、発光制御処理負荷を軽減することができる。
【０２７５】
また、実施形態の遊技機は、少なくとも第一発光素子群を駆動する第一発光駆動手段（例えば、一つのＬＥＤドライバ９０）と、少なくとも第二発光素子群を駆動する第二発光駆動手段（例えば、上記一つのＬＥＤドライバ９０とは別のＬＥＤドライバ９０）と、第一発光駆動手段と第二発光駆動手段とを制御する演出制御手段（演出制御ＣＰＵ２００）と、を備えている。
そして、演出制御手段は、第一発光素子群の発光パターン情報（発光パターン識別子）である第一発光パターン情報と、前記第二発光素子群の発光パターン情報である第二発光パターン情報とが少なくとも格納された発光パターンテーブル（ランプデータテーブル：図３４参照）と、複数の点灯データが格納された点灯データテーブル（変換テーブル：図３５参照）と、を有し、第一発光パターン情報に基づいて、点灯データテーブルに格納された複数の点灯データから何れかの点灯データを選択し、選択した点灯データにより第一発光素子群を発光可能であり、第二発光パターン情報に基づいて、点灯データテーブルに格納された複数の点灯データから何れかの点灯データを選択し、選択した点灯データにより第二発光素子群を発光可能であり、発光パターンテーブルにおいて、少なくとも第一発光パターン情報と、第二発光パターン情報とで、共通の発光パターン情報である特定発光パターン情報を記憶し、特定発光パターン情報に基づいて、点灯データテーブルに格納された特定の点灯データを選択可能としたものである。
【０２７６】
これにより、点灯データの重複格納量を抑制することが可能とされる。
従って、発光制御に要するメモリ消費量の削減を図ることができる。
【０２７７】
さらに、実施形態の遊技機においては、発光パターンテーブルは、所定単位の時間情報（タイマ（frame））に対して、少なくとも第一発光パターン情報及び／又は第二発光パターン情報が格納されている。
【０２７８】
これにより、発光パターンテーブルにおける同一の発光パターン情報の重複格納量が抑制される。
従って、メモリ消費量のさらなる削減を図ることができる。
【０２７９】
さらにまた、実施形態の遊技機においては、点灯データテーブルには、点灯データとして、第１ビット数（例えば８ビット）より少ないビット数である第２ビット数（例えば４ビット）による情報が格納され、演出制御手段は、点灯データテーブルから選択した第２ビット数による点灯データを第１ビット数に変換して各発光素子の駆動データを生成し、該駆動データを少なくとも第一、第二発光駆動手段に出力している（ステップＳ８１０）

。

【０２８０】

これにより、点灯データのデータ容量圧縮が図られる。

従って、メモリ消費量のさらなる削減を図ることができる。

【０２８１】

また、実施形態の遊技機においては、発光素子群を区分する単位がドライバ（ＬＥＤドライバ９０）単位とされている。

【０２８２】

発光素子群を区分する単位がドライバ単位とされることで、発光パターンテーブルの総データ容量の圧縮効果が高まり、メモリ消費量の削減を図ることができる。

【０２８３】

また、実施形態の遊技機においては、発光色が異なる複数の発光素子で成る発光ユニットを複数有し、発光素子群を区分する単位が発光ユニット単位とされている。

【０２８４】

発光素子群を区分する単位が発光ユニット単位とされることで、変換テーブルの総データ容量の圧縮効果が高まり、メモリ消費量の削減を図ることができる。

【０２８５】

また、実施形態の遊技機は、次のような構成であると換言することもできる。

すなわち、複数の発光素子（ＬＥＤ１２０）を有する発光手段（発光部２０ｂ、２０ｗ）と、発光素子を発光駆動する発光駆動手段（ＬＥＤドライバ９０）と、実行すべき演出動作の進行内容を表す演出進行データに基づき発光駆動手段を制御する演出制御手段（演出制御ＣＰＵ２００）と、を備えている。

そして、演出進行データに基づき特定され得る発光動作の種類ごとに、発光手段における発光素子群ごとの発光パターン情報が格納された発光パターンテーブル（ランプデータテーブル）と、発光パターン情報ごとに、発光素子群における各発光素子の発光輝度情報（点灯データ）が格納された点灯データテーブル（変換テーブル）と、が記憶されている。

その上で、演出制御手段は、演出進行データに基づき実行すべき発光動作が特定された場合に、発光パターンテーブルにおける該特定された発光動作に対応する発光素子群ごとの発光パターン情報を点灯データテーブルに基づいて点灯データに変換する変換手段（ステップＳ９０１）と、変換手段により得られた点灯データに基づいて発光駆動手段により発光素子を発光させる発光制御手段（ステップＳ８１０）と、を有している。

これにより、点灯データの重複格納量を抑制することが可能とされ、発光制御に要するメモリ消費量の削減を図ることができる。

【０２８６】

なお、第一実施形態については、上記で挙げた例に限らず多様な変形例や適用例が考えられる。

例えば上記では、それぞれの発光素子群における発光素子数を同一とする例を挙げたが、発光素子群間で発光素子数を一致させることは必須ではない。発光素子群間で発光素子数を異ならせた場合、発光素子数の異なる発光素子群ごとに、対応する変換テーブルを設ける。例えば、発光素子数が「１０」である第一種の発光素子群、発光素子数が「３０」である第二種の発光素子群、発光素子数が「５０」の第三種の発光素子群が存在する場合、変換テーブルは、これら第一～第三種の発光素子群ごとに設ける。

【０２８７】

また上記では、ＬＥＤによるランプ部６３，６４としては枠側に１系統（１つのシリアルデータ出力チャネル）、盤側に１系統の例を挙げたが、もちろんこれに限られない。枠側のランプ部６３について複数系統を設けても良いし、盤側のランプ部６４において複数系統を設けても良い。

また１つの系統に盤側の発光部２０ｂと枠側の発光部２０ｗが混在していてもよい。

【０２８８】

また、第一実施形態に係る技術については、弾球遊技機に限らず回胴式遊技機（いわゆるスロット機）にも適用できる。
例えばＬＥＤ駆動データのシリアルデータ送信は、スロット機の筐体前側の装飾ＬＥＤとリールのバックライトＬＥＤのそれぞれを第１系統、第２系統として行うことが考えられる。或いは、筐体前側の上部と下部をそれぞれシリアルデータ送信の系統に分けても良い。
【０２８９】
また、発光駆動の対象とする発光素子としてはＬＥＤに限定されず、例えば白熱電球等の他の発光素子とすることもできる。
【０２９０】
＜２．第二実施形態＞
［2-1. 第二実施形態のパチンコ遊技機の制御構成］

続いて、第二実施形態としてのパチンコ遊技機１Ａについて説明する。
第二実施形態のパチンコ遊技機１Ａは、演出制御用のコンピュータ装置を１ＣＰＵ化したものである。
なお、パチンコ遊技機１Ａの外観については図１、２で説明したものと略同様となることから説明は省略する。以下の説明において、既に説明済みとなった部分と同様となる部分については同一符号を付して説明を省略する。
【０２９１】
図３８は、パチンコ遊技機１Ａの内部構成の概略的なブロック図である。
先の図３と対比して分かるように、パチンコ遊技機１Ａは、パチンコ遊技機１と比較して、液晶制御基板５２と液晶インターフェース基板６６が設けられていない点が異なる。
また、本例では、音源ＩＣ５９も省略された構成とされている。
【０２９２】
パチンコ遊技機１Ａにおいては、演出制御部５１に代えて演出制御部５１Ａが設けられている。演出制御部５１Ａは、いわゆる１ＣＰＵとして、第一実施形態の演出制御部５１が担っていた機能と共に、液晶制御基板５２（前述した液晶制御ＣＰＵ）が担っていた機能を実現するように構成されている。
また本例における演出制御部５１Ａは、液晶制御ＣＰＵの機能と共に、音源ＩＣ５９に相当する機能も実現するように構成されている（後述する音コントローラ２３０）。
【０２９３】
なお、本例では、液晶表示装置３２として、主液晶表示装置３２Ｍと副液晶表示装置３２Ｓとが設けられた構成とされているが、第一実施形態と同様に単一の液晶表示装置３２のみを備えた構成とすることもできる。
主液晶表示装置３２Ｍでは、後述する演出制御部５１の制御の下、背景画像上で例えば左、中、右の３つの装飾図柄の変動表示が行われる。また通常演出、リーチ演出、スーパーリーチ演出などの各種の演出画像の表示も行われる。副液晶表示装置３２Ｓも、同様に各種演出に応じた表示が行われる。
なお、主液晶表示装置３２Ｍは、図柄表示部３３による第１、第２特別図柄の変動表示と時間的に同調して、画像による装飾図柄を変動表示する。
【０２９４】
演出制御部５１Ａは、主液晶表示装置３２Ｍ、副液晶表示装置３２Ｓに対する制御装置としての機能も備えているため、いわゆるＶＤＰ（Video Display Processor）、画像ＲＯＭ、ＶＲＡＭ（Video RAM）としての機能も備えられている。また、この場合の演出制御ＣＰＵ２００は、液晶制御部としても機能する。
ＶＤＰは、画像展開処理や画像の描画などの映像出力処理全般の制御を行う機能を指している。
画像ＲＯＭとは、ＶＤＰが画像展開処理を行う画像データ（演出画像データ）が格納されているメモリを指す。

ＶＲＡＭは、ＶＤＰが展開した画像データを一時的に記憶する画像メモリ領域である。
【０２９５】
演出制御部５１Ａは、これらの構成により、主制御部５０からのコマンド（演出制御コマンド）に基づいて各種の画像データを生成し、主液晶表示装置３２Ｍや副液晶表示装置３２Ｓに出力する。これによって主液晶表示装置３２Ｍや副液晶表示装置３２Ｓにおいて各種の演出画像が表示される。
【０２９６】
［2-2. 演出制御部の構成］

演出制御部５１Ａの具体的な構成例を図３９に示す。
図３９の例の演出制御部５１Ａは、例えば1チップマイクロコンピュータ２５０に対して演出制御ＲＯＭ２０１、Ｄ－ＲＡＭ２０２ａ、ＣＧ－ＲＯＭ２０６、WDT（ウォッチドッグタイマ）回路２１０等が外付け接続されて構成されている。
【０２９７】
１チップマイクロコンピュータ２５０は、演出制御ＣＰＵ２００を含むとともに、図示する各部によって、上述したＶＤＰ、ＶＲＡＭ等の機能や、さらには発光制御、モータ制御、音制御を行う機能を有する。
構成各部について説明していく。
【０２９８】
１チップマイクロコンピュータ２５０は演出制御ＣＰＵ２００を搭載している。演出制御ＣＰＵ２００は上述したように各種の制御処理を行う。
演出制御ＣＰＵ２００は、ＣＰＵインターフェース２０３を経由してマイクロコンピュータ２５０の内部デバイスや外部デバイスを使用する。
即ち演出制御ＣＰＵ２００は、ＣＰＵインターフェース２０３を経由してホストインターフェース２０４に接続され、ホストインターフェース２０４を介して、主制御部５０からのコマンド受信、演出制御ＲＯＭ２０１に対する読出アクセス、ＷＤＴ回路２１０との信号送受信、及びバス２５１を介した内部の各部との通信を行う。
【０２９９】
演出制御ＣＰＵ２００は、上述した演出制御ＲＡＭ２０２として、Ｄ－ＲＡＭ２０２ａ及び内蔵ＣＰＵ用ワークメモリ２０２ｂを用いる。
Ｄ－ＲＡＭ２０２ａは、ＲＡＭインターフェース２０８に接続されており、演出制御ＣＰＵ２００は、ＲＡＭインターフェース２０８を介してＤ－ＲＡＭ２０２ａに対する書き込みや読み出しを行う。
【０３００】
システム制御レジスタ２０５はシステムの初期設定をするためのレジスタである。
転送回路２１２は、マイクロコンピュータ２５０の内部リソースや外部ペリフェラルを入出力とした転送を行う。
【０３０１】
ＣＧ－ＲＯＭ２０６は画像ＲＯＭであり、表示するキャラクタ画像データを格納する。
ＣＧ－ＲＯＭ２０６はマイクロコンピュータ２５０に設けられたＣＧバスインターフェース２０７に接続されている。これによりマイクロコンピュータ２５０内の所要部位は、ＣＧバスインターフェース２０７を介してＣＧ－ＲＯＭ２０６にアクセスできる。
【０３０２】
マイクロコンピュータ２５０は例えば容量が４８ＭバイトのＶＲＡＭ２０９を内蔵している。ＶＲＡＭ２０９は描画素材やフレームデータを格納する。
ＶＲＡＭ２０９の使用態様は設定により各種可能であるが、本例では表示用のフレームデータの記憶領域として２つのフレームバッファ２０９Ａ、２０９Ｂを設定し、各フレームバッファ２０９Ａ、２０９Ｂを１フレーム毎に交互に用いて描画や表示データ出力を行うようにする。
【０３０３】

なお、本例では、２つの表示装置（主液晶表示装置３２Ｍ、副液晶表示装置３２Ｓ）を使用するので、フレームバッファ２０９Ａ、２０９Ｂとしては主液晶表示装置３２Ｍ、副液晶表示装置３２Ｓのそれぞれに対応して用意される。

つまり図示するフレームバッファ２０９Ａは、主液晶表示装置３２Ｍ用のフレームバッファ領域と副液晶表示装置３２Ｓ用のフレームバッファ領域を含み、またフレームバッファ２０９Ｂは、主液晶表示装置３２Ｍ用のフレームバッファ領域と副液晶表示装置３２Ｓ用のフレームバッファ領域を含む。

ＶＲＡＭ２０９のフレームバッファ２０９Ａ、２０９Ｂにはそれぞれ、主液晶表示装置３２Ｍ、副液晶表示装置３２Ｓ用の１フレーム分の画像データが各々生成されるが、所定の描画コマンドによって、フレームバッファ２０９Ａ、２０９Ｂの描画位置が特定される。

【０３０４】

インデックステーブル２１１は、ＶＲＡＭ２０９の記憶領域を設定するなどＶＲＡＭ２０９の管理を行う。

【０３０５】

プリローダ２２０、表示回路２２１，２２２，２２３、ＧＤＥＣ（グラフィックデコーダ）２２３、描画回路２２５は、ビデオプロセッサ（ＶＤＰ）としての各種処理を行う。

演出制御ＣＰＵ２００は、演出制御コマンドに応じて主液晶表示装置３２Ｍや副液晶表示装置３２Ｓで表示する画像内容を設定するディスプレイリストを作成する。このディスプレイリストとは、描画コマンドの集まりのことである。

ディスプレイリストは、描画する順番に記載された一群の描画コマンドで構成されている。描画コマンドには、１フレームのどの位置に、どのような画像を描画するかを規定するコマンドも含まれ、描画すべき画像のＣＧ－ＲＯＭ２０６などの記憶位置（ソースアドレス）も特定されている。

【０３０６】

描画回路２２５は、ＶＲＡＭ２０９上のフレームバッファ（２０９Ａ又は２０９Ｂ）に図形を描画する。

ＧＤＥＣ２２３は、描画で使用する圧縮画像データをデコードしＶＲＡＭ２０９に展開する。

【０３０７】

プリローダ２２０は転送回路２１２によって送信されたディスプレイリストを解析し、その中で参照している画像素材、即ちＣＧ－ＲＯＭ２０６上の画像データをＤＲＡＭ２０２ａまたはＶＲＡＭ２０９のいずれかの指定されている領域に転送する。

またこのときプリローダ２２０は、画像データの参照先を、転送後のアドレスに書換えたディスプレイリストを出力する。書換えられたディスプレイリストは、転送回路２１２によって描画回路２２５に送信される。

【０３０８】

描画回路２２５による描画の実行中は、画像データへのランダムアクセスが発生するため、例えばＮＡＮＤフラッシュメモリのようなランダムアクセスに弱いＣＧ－ＲＯＭ２０６から画像データを直接参照すると描画性能が大きく低下する。そこでプリローダ２２０を使って、あらかじめ画像データやディスプレイリストをＤ－ＲＡＭ２０２ａ、または内蔵のＶＲＡＭ２０９に転送しておくことで、高速に動画デコード、および描画を実行することができるようにしている。

【０３０９】

上記の通り本実施形態では、プリローダ２２０を機能させているので、ディスプレイリストの画像データ（画像素材としてのＣＧデータ）の参照先は、ＣＧ－ＲＯＭ２０６ではなく、Ｄ－ＲＡＭ２０２ａ又はＶＲＡＭ２０９に設定されたプリロード領域である。そのため、描画回路２２５による描画の実行中に生じる画像データへのランダムアクセスを迅速に実行することができ、動きの激しい高解像度の動画についても描画処理に有利である。

【０３１０】
表示回路２２１，２２２，２２３は、ＶＲＡＭ２０９のフレームバッファ（２０９Ａ又は２０９Ｂ）の表示データを外部に表示出力する。
ここでマイクロコンピュータ２５０が３つの表示回路２２１，２２２，２２３を備えているのは、１系統のデジタルＲＧＢ出力と、２系統のＬＶＤＳ（Low voltage differential signaling）出力を可能とする３画面出力対応の構成を採っているためである。本例の場合、例えばデジタルＲＧＢ出力を主液晶表示装置３２Ｍへの表示データ出力として用い、一つのＬＶＤＳ出力を副液晶表示装置３２Ｓへの表示データ出力として用いる。
表示回路２２１，２２２，２２３からの表示データは、ＬＣＤインターフェース２２６によって選択されて外部機器、即ち主液晶表示装置３２Ｍと副液晶表示装置３２Ｓに出力される。
【０３１１】
表示回路２２１，２２２，２２３は、ノンインタレース出力としての表示データ出力を行う。また表示データについて、例えば２倍から１／２倍までの拡大・縮小を行うスケーリング処理機能や、ディザリング機能、カラー補正機能を備えている。
【０３１２】
図４０は、表示回路２２１，２２２，２２３及びＬＣＤインターフェース２２６のブロック図を示している。
各表示回路２２１，２２２，２２３は同期信号生成部２６５を有し、それぞれ画像のフレームに同期して処理を行う。同期信号生成部２６５は、同期信号として、例えば垂直ブランク期間（非有効ライン走査期間）を示すＶブランク信号、水平同期信号、垂直同期信号を生成する。これらの同期信号は、マイクロコンピュータ２５０の内部での制御処理に用いることができるほか、主液晶表示装置３２Ｍ、副液晶表示装置３２Ｓへの表示データ転送や表示制御動作にも用いられる。
なお、各表示回路２２１，２２２，２２３で用いる同期信号同士は同期させることができる。例えば主液晶表示装置３２Ｍなどの外部ディスプレイの垂直同期信号の立ち下がりをトリガにして、表示回路２２１，２２２，２２３を強制的に同期させることができる。
【０３１３】
各表示回路２２１，２２２，２２３は、ＶＲＡＭ読出部２６１により、ＶＲＡＭ２１０のフレームバッファ２０９Ａ又は２０９Ｂから読み出したフレームデータを取得する。
表示回路２２１，２２２については、取得したフレームデータについてスケーラ２６２でのスケーリング処理、カラー補正部２６３でのカラー補正処理、ディザラー２６４でのディザリング処理を施すことが可能とされる。そしてディザラー２６４による処理の後の出力としてデジタルＲＧＢ出力とＬＶＤＳ出力が行われる。表示回路２２３も同様であるが、この図の例では表示回路２２３はスケーラ２６２の機能を搭載していない例を挙げた。
【０３１４】
ＬＣＤインターフェース２２６は、デジタルＲＧＢ出力部２２６ａとＬＶＤＳ出力部２２６ｂを有する。
デジタルＲＧＢ出力部２２６ａには、表示回路２２１，２２２，２２３からのデジタルＲＧＢ出力としての表示データが供給される。デジタルＲＧＢ出力部２２６ａは、表示回路２２１，２２２，２２３の１つのデジタルＲＧＢ出力を選択して、主液晶表示装置３２Ｍに供給する。
またＬＶＤＳ出力部２２６ｂには、表示回路２２１，２２２，２２３からのＬＶＤＳ出力としての表示データが供給される。ＬＶＤＳ出力部２２６ｂは、表示回路２２１，２２２，２２３のうちで２つのＬＶＤＳ出力を選択して出力することができる。本実施形態のパチンコ遊技機１の場合、表示回路２２１，２２２，２２３のうちの１つのＬＶＤＳ出力を副液晶表示装置３２Ｓに供給する。
もちろんこれは一例であり、１つのＬＶＤＳ出力を主液晶表示装置３２Ｍに供給することも考えられるし、デジタルＲＧＢ出力を副液晶表示装置３２Ｓに供給することも考えら

れる。
【０３１５】
図３９のマイクロコンピュータ２５０は音コントローラ２３０を有する。
音コントローラ２３０は、例えば音源制御部と圧縮音声データを記憶する音データ記憶部を有し、演出制御ＣＰＵ２００の制御に基づいて、圧縮音声データをデコードして出力音声信号を生成し、外部のアンプ部６７へ出力する。
なお、音コントローラ２３０の搭載の有無にかかわらず、演出制御ＣＰＵ２００はマイクロコンピュータ２５０外部の音源ＩＣを制御して、音声信号をアンプ部６７へ出力するようにしてもよい。
【０３１６】
マイクロコンピュータ２５０は汎用ポート２３１を有している。例えば８ビットの入出力ポートとされる。汎用ポート２３１により各種の外部デバイスとの入出力が可能となる。
【０３１７】
シリアル出力コントローラ２４０は、制御信号をシリアルデータとして外部に出力する。本実施形態の場合、シリアル出力コントローラ２４０にはランプコントローラ２４１とモータコントローラ２４２が設けられ、ランプ部６３，６４の発光駆動データの出力や可動体役物モータ６５のモータ駆動データの出力が行われる。
【０３１８】
なお、枠ドライバ部６１、盤ドライバ部６２、モータドライバ部７０の構成については先の図４や図５で説明したものと同様となることから重複説明は避ける。
【０３１９】
演出制御部５１Ａは、パラレル／シリアル変換部２６０を有している。各可動体役物モータ６５（ステッピングモータ１２１）のそれぞれに対しては、図３で説明した原点スイッチ６８が対応して配置されており、原点スイッチ６８からの信号はパラレル／シリアル変換部２６０に入力される。
また図３９では、操作部６０からのユーザの操作情報もパラレル／シリアル変換部２６０に入力される例を示している。ユーザの操作情報とは、演出ボタン１１，１２、十字キー１３等の操作に応じた信号である。
パラレル／シリアル変換部２６０は、例えば１チップマイクロコンピュータ２５０とは別体のＩＣとして演出制御部（演出制御基板）５１Ａに搭載されるが、１チップマイクロコンピュータ２５０に内蔵されてもよい。
【０３２０】
パラレル／シリアル変換部２６０は、例えば３２系統の信号を入力し、入力したデータをシリアルデータIsに変換してモータコントローラ２４２に供給する。
パラレル／シリアル変換部２６０の動作はモータコントローラ２４２からの制御信号CNTによって制御される。パラレル／シリアル変換部２６０はクロック信号CLKを用いてシリアルデータ転送を行う。
この構成により、シリアル出力コントローラ２４０が、各可動体役物モータ６５の原点検出状態と演出スイッチ１１，１２等の操作部６０に対するユーザ操作をまとめて、つまり演出制御に必要な入力をまとめて効率的に検出できる。従って演出制御ＣＰＵ２００は効率よく演出状態を把握できる。
【０３２１】
［2-3. 演出制御部の機能対比］

図４１は、第一実施形態のように演出制御部５１としてのコンピュータ装置とは別途に液晶制御のためのコンピュータ装置（液晶制御基板５２）を設けた構成（以下「２ＣＰＵ構成」とも表記する）とした場合における演出制御部５１、液晶制御基板５２の各機能構成を模式的に表した機能ブロック図である。
先の第一実施形態の説明から理解されるように、演出制御部５１は、主制御部５０から

の演出制御コマンドを受信・解析し、解析結果に基づいてランプ、音、モータとしての画像演出以外の演出についてシナリオデータ（演出進行データ：具体的には図２０や図２１に示したような各種の登録情報）を生成し、該シナリオデータに基づきランプ、音、モータについての駆動データを生成・出力して所望の演出動作を実現している。

　また、液晶に関しては、演出制御コマンドの解析結果に基づき得られる液晶制御コマンドを液晶制御基板５２側に送信する。

【０３２２】

　演出制御部５１は、主制御部５０が送信した演出制御コマンドを保持するためのバッファ領域（記憶領域）である演出制御コマンドバッファＦ１１（前述した「コマンド受信バッファ」）を有すると共に、演出制御コマンドバッファＦ１１に保持された演出制御コマンドについての解析を行う演出制御コマンド解析処理部Ｆ１２と、演出制御コマンド解析処理部Ｆ１２によるコマンド解析結果に基づきランプ、音、モータについてのシナリオデータを生成し、生成したシナリオデータに基づく処理を実行する演出制御シナリオ実行処理部Ｆ１３と、液晶制御コマンドの出力バッファであるコマンド出力バッファＦ１４とを有している。

【０３２３】

　演出制御コマンド解析処理部Ｆ１２では、演出制御コマンドの解析処理として、先の図１３に示した処理を行い、コマンドの種類を特定する。演出制御コマンドは、その種類ごとに実行すべき処理が予め定められている。演出制御コマンドの種類によっては、液晶制御基板５２に対する所定の液晶制御コマンドの送信処理を行うことが定められたものある。例えば、図１５Ａや図１５Ｂで例示した「コマンドセット」の処理等である。この場合、演出制御コマンド解析処理部Ｆ１２の解析処理の結果として、所定の液晶制御コマンドがコマンド出力バッファＦ１４にセットされる。

　また、演出制御コマンドとしては、液晶制御基板Ｆ１２に対して当該演出制御コマンドと同一のコマンドを送信する（つまりスルー送信する）処理を行うべきことが定められたものもある。この場合も、演出制御コマンド解析処理部Ｆ１２の解析処理の結果として、液晶制御コマンド（データとしては受信した演出制御コマンドと同一）がコマンド出力バッファＦ１４にセットされる。

　なお、コマンドのスルー送信については後の図４７～図４９においても説明する。

【０３２４】

　演出制御シナリオ実行処理部Ｆ１３において、コマンド解析結果に基づくシナリオデータの生成は、例えば図１４ＤのステップＳ３４１のようにコマンド対応処理（Ｓ３０６）の一部として設けられたシナリオ登録の処理によって実現される。

　また、演出制御シナリオ実行処理部Ｆ１３が実行するシナリオデータに基づく具体的な処理としては、図１８、図１９に示したシナリオ更新処理、サブシナリオの更新処理や、図３６、図２５に示したＬＥＤ駆動データ更新処理、ＬＥＤ駆動データ出力処理等を挙げることができる。

【０３２５】

　ここで、演出制御コマンドとしては、当該演出制御コマンドの受信から所定時間の経過後に液晶制御基板５２に対する所定の液晶制御コマンドの送信を行うべきことが定められたものもある（以下、液晶制御コマンドの「時間差送信」とも表記する）。

　第一実施形態では説明を省略したが、本例では、このような時間差送信を行う場合には、液晶制御コマンドについて送信対象のコマンドと送信タイミングとを管理するためのシナリオデータ（以下「コマンド送信シナリオデータ」とも表記する）が用いられる。

　演出制御コマンド解析処理部Ｆ１２によるコマンド解析の結果、このようなコマンド送信シナリオデータが生成された場合には、演出制御シナリオ実行処理部Ｆ１３が該コマンド送信シナリオデータに従った処理を実行することで、液晶制御基板５２への液晶制御コマンドの送信が行われる。つまりこの場合は、演出制御シナリオ実行処理部Ｆ１３によるシナリオ実行に伴い所定の液晶制御コマンドがコマンド出力バッファＦ１４にセットされる。

液晶制御コマンドの時間差送信については後の図４７～図４９においても説明する。

【０３２６】

なお、コマンド出力バッファＦ１４にセットされた液晶制御コマンドはコマンド出力処理（図１２のステップＳ２０６）によって液晶制御基板５２に送信される。

【０３２７】

液晶制御基板５２は、演出制御部５１が送信した液晶制御コマンドを保持するためのバッファ領域である液晶制御コマンドバッファＦ２１と、液晶制御コマンドバッファＦ１１に保持された液晶制御コマンドについての解析を行う液晶制御コマンド解析処理部Ｆ２２と、液晶制御コマンド解析処理部Ｆ２２によるコマンド解析結果に基づき液晶演出についてのシナリオデータ（以下「液晶シナリオデータ」とも表記する）を生成し、生成したシナリオデータに基づく処理を実行する液晶制御シナリオ実行処理部Ｆ２３とを有している。

液晶制御シナリオ実行処理部Ｆ２３が液晶シナリオデータに基づく処理を行うことで、前述したディスプレイリストの作成や描画が行われて画像演出を実現する上で必要な画像データが適宜生成され、該画像データが液晶表示装置３２に出力される。これにより、液晶表示装置３２に各種の演出画像が表示される。

【０３２８】

ここで、上記のような液晶制御基板５２が有する画像演出についての制御機能を演出制御部５１に実装させて１ＣＰＵ化を図るとした場合には、画像演出以外の演出制御用のプログラム（主制御部５０から送信される演出制御コマンドの解析やランプ、音、モータについてのシナリオデータの生成など）と、画像演出制御用のプログラムとを統合した新たなプログラムを作成し、該プログラムを１ＣＰＵとしてのコンピュータ装置（第二実施形態における演出制御部５１Ａ）に実行させることが考えられる。

この際、１ＣＰＵとした場合には液晶制御基板５２に液晶制御コマンドを送信する必要はなくなるため、液晶シナリオデータの生成を、主制御部５０からのコマンドとは別途のコマンドを介在させずに行うようなプログラムを構成することが考えられる。

【０３２９】

しかしながら、これによると、１ＣＰＵに実行させる画像演出制御用のプログラムは、液晶制御基板５２（液晶制御ＣＰＵ）に実行させていたプログラムから比較的大きく変更することを要し、プログラム作成の作業負担が増大する虞がある。

特に、演出制御部５１のプログラムと液晶制御基板５２のプログラムとを別業者に作成させていた等の事情の下では、１ＣＰＵに実行させるべきプログラムの作成作業負担がより大きくなる。

【０３３０】

そこで第二実施形態では、１ＣＰＵとしての演出制御部５１Ａについて、図４２に示すような制御構成を採る。

図示のように演出制御部５１Ａは、演出制御部５１と同様に演出制御コマンドバッファＦ１１、演出制御コマンド解析処理部Ｆ１２、演出制御シナリオ実行処理部Ｆ１３を有している。この場合、コマンド出力バッファＦ１４は省略され、液晶制御コマンドバッファＦ１５が設けられる。さらに、演出制御部５１Ａは、液晶制御コマンド解析処理部Ｆ１６と液晶制御シナリオ実行処理部Ｆ１７とを有している。

液晶制御コマンド解析処理部Ｆ１６、液晶制御シナリオ実行処理部Ｆ１７の処理内容は、液晶制御基板５２における液晶制御コマンド解析処理部Ｆ２２、液晶制御シナリオ実行処理部Ｆ２３とそれぞれ同様である。すなわち、液晶制御コマンド解析処理部Ｆ１６は、液晶制御コマンドバッファＦ１５に保持された液晶制御コマンドについての解析を行い、液晶制御シナリオ実行処理部Ｆ１７は、液晶制御コマンド解析処理部Ｆ１６によるコマンド解析結果に基づき液晶シナリオデータを生成し、生成した液晶シナリオデータに基づく処理を実行することで液晶演出を実現する上で必要な画像データを生成し、液晶表示装置３２に表示させる。

【０３３１】

演出制御部５１Ａにおいて、演出制御コマンドバッファＦ１１は、演出制御ＲＡＭ２０２（例えばＤ－ＲＡＭ２０２ａ：図３９参照）の一部の記憶領域として設定されている。また、液晶制御コマンドバッファＦ１５としても、演出制御ＲＡＭ２０２の一部の記憶領域として設定されている。

演出制御部５１Ａの演出制御ＣＰＵ２００は、演出制御コマンド解析処理部Ｆ１２や演出制御シナリオ実行処理部Ｆ１３の処理に伴い液晶制御コマンドを送信すべきとされた場合には、対象の液晶制御コマンドを液晶制御コマンドバッファＦ１５にセットする。

【０３３２】

上記構成のように、演出制御部５１Ａは、主制御部５０により送信された演出制御コマンドを保持する演出制御コマンドバッファＦ１１と、演出制御コマンドを解析して得られる液晶制御コマンドを保持する液晶制御コマンドバッファＦ１５とを有している。

これにより、演出制御部５１Ａは、演出制御コマンドを解析し、解析により得た液晶制御コマンドを所定記憶領域にセットするという処理については、演出制御部５１の処理（コマンド出力バッファＦ１４に液晶制御コマンドをセットする処理）を踏襲することが可能とされる。

また、画像演出の制御処理についても、液晶制御コマンドバッファＦ１５にセットされた液晶制御コマンドを取得し、液晶制御コマンドに基づいた液晶シナリオデータの生成、及び液晶シナリオデータに基づいた液晶表示装置３２の制御（第二実施形態では主液晶表示装置３２Ｍと副液晶表示装置３２Ｓの制御）という、液晶制御基板５２と同様の処理を踏襲することが可能とされる。

これにより、１ＣＰＵとしての演出制御部５１Ａが実行すべきプログラムとしては、２ＣＰＵ時の演出制御部５１が実行していたプログラムと液晶制御基板５２（液晶制御ＣＰＵ）が実行していたプログラムをほぼそのまま踏襲することが可能とされる。具体的な変更点としては、主には液晶制御コマンドをコマンド出力バッファＦ１４ではなく液晶制御コマンドバッファＦ１５にセットする点である。

従って、１ＣＰＵを実現するにあたって既存プログラムの書き替えをほぼ不要とすることができ、プログラム作成に係る作業負担軽減を図ることができる。

【０３３３】

また、演出制御部５１Ａは、主制御部５０より送信された演出制御コマンドの解析結果に基づいて、画像表示以外の演出の進行を管理する第一演出進行管理手段（演出制御シナリオ実行処理部Ｆ１３）と、第一演出進行管理手段による演出の進行に応じて、画像表示に関する画像表示指示情報（液晶制御コマンド）を生成する画像表示指示情報生成手段（演出制御シナリオ実行処理部Ｆ１３）と、を有すると共に、画像表示指示情報生成手段により生成された画像表示指示情報に基づいて、画像表示に関する演出の進行を管理する第二演出進行管理手段（液晶制御コマンド解析処理部Ｆ１６及び液晶制御シナリオ実行処理部Ｆ１７）を有している。

第一演出進行管理手段、及び画像表示指示情報生成手段による処理は、演出制御部５１が有する演出制御シナリオ実行処理部Ｆ１３の処理と同様である。そして、第二演出進行管理手段による処理は、液晶制御基板５２が有する液晶制御コマンド解析処理部Ｆ２２及び液晶制御シナリオ実行処理部Ｆ２３の処理と同様である。

これにより、１ＣＰＵとしての演出制御部５１Ａが実行すべきプログラムとしては、２ＣＰＵ時の演出制御部５１が実行していたプログラムと液晶制御基板５２（液晶制御ＣＰＵ）が実行していたプログラムをほぼそのまま踏襲することが可能とされる。

従って、１ＣＰＵを実現するにあたって既存プログラムの書き替えをほぼ不要とすることができ、プログラム作成に係る作業負担軽減を図ることができる。

【０３３４】

ここで、上記の第一演出進行管理手段及び画像表示指示情報生成手段の処理を実現するためのプログラムと、第二演出進行管理手段の処理を実現するためのプログラムは、それぞれ独立したプログラムモジュールとして構成することができる。

【０３３５】

［2-4．第二実施形態の演出制御部の処理］

続いて、演出制御部５１Ａの処理について説明する。
演出制御部５１Ａの演出制御ＣＰＵ２００は、図１１、図１２に示した演出制御メイン処理、１ｍｓタイマ割込み処理を除く処理（図１３〜図１９、図２５、及び図３６）については、第一実施形態の演出制御ＣＰＵ２００と同様の処理を行う。以下では、主に演出制御部５１の演出制御ＣＰＵ２００が実行する処理と異なる部分について説明する。
【０３３６】
演出制御部５１Ａの演出制御ＣＰＵ２００は、図１１に示した演出制御メイン処理に代えて、図４３に示す演出制御メイン処理を実行する。
図１１に示したメイン処理との大きな相違点は、この場合の演出制御ＣＰＵ２００が１ＣＰＵとして液晶制御も行う関係から、表示データのフレームタイミングに基づいて処理を進行するという点である。
【０３３７】
図４３において、演出制御ＣＰＵ２００は、遊技機本体に対して電源が投入されると演出制御メイン処理を開始する。
演出制御メイン処理において、演出制御ＣＰＵ２００は、まずステップＳ１０００で、遊技動作開始前における必要な初期設定処理を行う。例えば初期設定処理として、インターフェース系の初期化、割込設定、外部メモリの初期化、ＷＤＴ初期化、可動体役物の起点復帰処理及び制御の初期化、音源制御初期化、シリアル出力コントローラ２４０の初期化、スケジューラ（シナリオスケジューラ、デバイススケジューラ、ランプスケジューラ、サウンドスケジューラ等）の初期化、システムタイマの初期化、液晶制御初期化等を行う。
なお各スケジューラは、上述のシナリオ登録情報、モータデータ登録情報、ランプデータ登録情報、音データ登録情報、及びコマンド送信シナリオデータのこと、もしくはこれらに基づいたシナリオ制御の進行を指す。初期化処理としては、これらシナリオ登録情報等を記憶するワーク領域を初期化する。
【０３３８】
続くステップＳ１００１で演出制御ＣＰＵ２００は、処理時間の計測開始処理を行う。この処理時間は、表示画像の各フレーム期間内において、フレーム開始から終了時点までの経過時間を把握するために計測されるものである。処理時間は、前述したＶブランク信号に基づき０リセットされる。具体的に本例では、Ｖブランク信号は１フレーム期間につき２回発生され（フレーム期間の前半終了時、後半終了時にそれぞれ１回発生され）、処理時間は２回目のＶブランク信号の発生に応じて０リセットされる。
【０３３９】
ステップＳ１００１で処理時間の計測を開始すると、演出制御ＣＰＵ２００はステップＳ１００２以降の処理を繰り返し行う。
処理の概要を説明すると、先ずは、フレーム期間の開始タイミングでステップＳ１００２〜Ｓ１０１０の処理が１回行われた上で、ステップＳ１０２１〜Ｓ１０２４による演出制御コマンドの解析に係る処理、ステップＳ１０２５〜Ｓ１０２８による液晶制御コマンドの解析に係る処理、ステップＳ１０３０〜Ｓ１０３４による各種スケジューラの実行に係る処理、及びステップＳ１０４０〜Ｓ１０４３によるランプ駆動データの作成・出力やキーイベントに係る処理が行われる。さらに、フレーム期間の終了時には、ステップＳ１０５０〜Ｓ１０５１による処理が実行される。
【０３４０】
本例では、フレーム期間の終了タイミング判定（Ｓ１０５０）には、表示制御に用いる同期信号であるＶブランク信号が用いられる。Ｖブランク信号等の表示データの同期信号は、１チップマイクロコンピュータ２５０内のＶＤＰ機能を実現する部位で生成される。例えば同期信号生成部２６５で生成される。本実施形態では１フレーム期間である１／３０秒（３３．３ｍｓ）の間において、Ｖブランク信号は２回発生される。すなわち、Ｖブ

ランク信号の周期は１／６０秒（１６．６ｍｓ）である。

【０３４１】

本例では、ステップＳ１０２１～Ｓ１０２４による演出制御コマンドの解析に係る処理、及びステップＳ１０２５～Ｓ１０２８による液晶制御コマンドの解析に係る処理は、フレーム期間開始時からの経過時間が１６ｍｓ以内である場合に実行するようにされている。また、ステップＳ１０４０～Ｓ１０４３によるランプ駆動データの作成・出力やキーイベントに係る処理は、フレーム期間開始からの経過時間が２０ｍｓ以内である場合に実行するようにされている。

上述した処理時間（Ｓ１００１）は、これらの時間管理を行うために計測されるものである。

【０３４２】

ステップＳ１００２で演出制御ＣＰＵ２００は、ＷＤＴ回路２１０に対するクリア制御を行う。従ってＷＤＴ回路２１０は演出制御ＣＰＵ２００が正常な処理状態であれば１フレーム毎にリセットされる。

ステップＳ１００３で演出制御ＣＰＵ２００は、フレーム更新フラグを確認する。フレーム更新フラグは、スケジューラ更新等をフレーム期間で管理するためのフラグである。表示データの或るフレームの開始時点ではフレーム更新フラグはオフとされている（フレーム終了時に後述するステップＳ１０５２でオフとするため）。

従ってフレーム開始タイミングでは、演出制御ＣＰＵ２００はステップＳ１００３からＳ１００４に進む。

【０３４３】

ステップＳ１００４で演出制御ＣＰＵ２００は、液晶制御フレーム更新処理を実行する。これは、前述した液晶シナリオデータによって進行管理が行われている画像演出についてのシナリオを、１フレーム分進める処理に相当する。

液晶シナリオデータには、図２０Ａに示したシナリオ登録情報の「待機時間（delay）」や「サブシナリオタイマ（scTm）」のように、シナリオの進行タイミングを管理する例えばタイマ値等による値が登録情報の一つに含まれている。従って、ステップＳ１００４の液晶制御フレーム更新処理では、該値を更新することで画像演出のシナリオを１フレーム分進める。

【０３４４】

ステップＳ１００４に続くステップＳ１００５で演出制御ＣＰＵ２００は、描画更新処理を行う。すなわち、液晶シナリオデータに基づき、例えば前述のように描画する順番に記載された一群の描画コマンドであるディスプレイリストの作成や、プリロードの実行開始制御を行う。

【０３４５】

ステップＳ１００５の描画更新処理を実行したことに応じ、演出制御ＣＰＵ２００はステップＳ１００６～Ｓ１００８の処理によりシナリオ、ランプ、サウンド（音）の各スケジューラのフレーム更新を行う。

具体的に、ステップＳ１００６のシナリオスケジューラ更新処理では、図２０Ａに示したシナリオ登録情報における待機時間（delay）、及びメインシナリオタイマ（msTm）の値を更新する。

また、ステップＳ１００７のランプスケジューラフレーム更新処理では、図２０Ｂに示したランプデータ登録情報における実行時間（time）の値を更新する。

さらに、ステップＳ１００８のサウンドスケジューラフレーム更新処理では、図２０Ａに示したシナリオ登録情報におけるサブシナリオタイマ（scTm）の値を更新する。なお、図１９Ａで説明したサブシナリオの更新処理を参照して分かるように、サブシナリオタイマ（scTm）の値は、音（及びモータ）の演出シナリオについての進行タイミングを管理するタイマ値として機能するものである。

【０３４６】

上記ステップＳ１００６～Ｓ１００８の各フレーム更新処理により、フレームの開始時

点において、メインシナリオの進行、ランプ演出のシナリオの進行、音演出のシナリオの進行がそれぞれ１タイミング進められることになる。

【０３４７】

演出制御ＣＰＵ２００は、ステップＳ１００６～Ｓ１００８の更新処理を実行したことに応じ、ステップＳ１００９でフレーム更新フラグをＯＮにする。これは、現在のフレームにおいてシナリオのタイマ進行を行ったことを示す情報となる。

また、演出制御ＣＰＵ２００は続くステップＳ１０１０で、スケジューラ更新フラグをＯＦＦとする。これは、スケジューラ更新、つまりタイマ進行に伴ったシナリオ内容（シナリオ登録情報、ランプデータ登録情報、音データ登録情報等）の更新がまだ行われていないことを示す情報となる。

【０３４８】

ステップＳ１０１０でスケジューラ更新フラグをＯＦＦとしたことに応じ、演出制御ＣＰＵ２００はステップＳ１００２に戻る。演出制御ＣＰＵ２００はＷＤＴ回路２１０のクリアを行った後、ステップＳ１００３ではフレーム更新フラグはＯＮになっているためステップＳ１０２０に進むことになる。

【０３４９】

なお、上記のようにフレーム開始タイミング後の最初の１回だけ、ステップＳ１００４～Ｓ１０１０の処理が行われ、そのうちのステップＳ１００６～Ｓ１００８では、画像演出以外の演出シナリオについてのタイマ更新が行われる。以降説明するように画像演出以外の演出についてはステップＳ１０３１～Ｓ１０３４等の処理によりシナリオの内容更新や制御が実行されるが、上記のようなタイマ更新により、この場合における演出のためのシナリオは、１フレーム期間である３３．３ｍｓ毎に進行することになる。例えばメインシナリオタイマ（msTm）は３３．３ｍｓ毎に進行する。このため、図２２のメインシナリオテーブルにおける１行のメインシナリオタイマ（msTm）の時間は、表記する値×３３．３ｍｓの時間に相当する。また図２３のサブシナリオテーブルの時間（time）で示す１行の時間も、表記する値×３３．３ｍｓの時間に相当する。

【０３５０】

この場合の演出制御メイン処理では、ステップＳ１０２０において、処理時間の値に応じて処理が分岐する。

演出制御ＣＰＵ２００は、ステップＳ１０２０において処理時間が３１ｍｓに達している、つまりフレーム期間の略終端に達しているか否かを判定し、３１ｍｓに達していればステップＳ１０２１に処理を進める。

演出制御ＣＰＵ２００は、処理時間が３１ｍｓに達しない限り、ステップＳ１０２１～Ｓ１０４３を可能な回数、繰り返すようにステップＳ１０２０からＳ１０２１に進む。

なお、３１ｍｓはあくまで一例でありこれに限定されるものではない。

【０３５１】

ステップＳ１０２１で演出制御ＣＰＵ２００は、処理時間が１６ｍｓ未満か否かで処理を分岐する。つまり現在が或るフレーム期間の前半に該当するか後半に該当するかである。なお、１６ｍｓはあくまで一例であり、これに限定されない。例えば、より正確に１６．６ｍｓとしてもよい。

現在、処理時間が１６ｍｓ未満である、つまりフレーム前半期間であれば、演出制御ＣＰＵ２００はステップＳ１０２２に進んで、演出制御コマンドの受信を確認する。すなわち、主制御部５０からの受信コマンドを確認する。具体的には、先に説明した演出制御コマンドバッファＦ１１（本例ではＤ－ＲＡＭ２０２ａの所定領域）に１以上の受信コマンドが記憶されているか否かを確認する。受信コマンドがなければ、演出制御ＣＰＵ２００はステップＳ１０２５に進む。

【０３５２】

一方、１又は複数の受信コマンドがあれば、演出制御ＣＰＵ２００はステップＳ１０２３の液晶制御コマンド解析処理として、受信コマンドについての解析処理を行う。この解析処理としては、図１３で説明したものと同様の処理を行い、従ってこの場合も図１４～

図１５で一部を例示したような各種受信コマンドに応じた処理が行われる。

【０３５３】

なお、図示は省略するが、主制御部５０が送信する演出制御コマンドについての受信処理は、演出制御部５１Ａ（演出制御ＣＰＵ２００）による外部割込み処理において実行される。

【０３５４】

ステップＳ１０２３に続くステップＳ１０２４で演出制御ＣＰＵ２００は、スケジューラ更新フラグをＯＦＦにする。このステップＳ１０２４でスケジューラ更新フラグをＯＦＦとすることは、現在のフレーム期間中において、以前に一旦ステップＳ１０３１でスケジューラ更新フラグをＯＮにした後であっても、コマンド受信があった場合は再びＯＦＦにするという意味を持つ。

【０３５５】

ステップＳ１０２５で演出制御ＣＰＵ２００は、先のステップＳ１０２１と同様に処理時間が１６ｍｓ未満であるか否かを判別し、１６ｍｓ未満であればステップＳ１０２６に進んで液晶制御コマンドの受信を確認する。具体的には、先に説明した液晶制御コマンドバッファＦ１５（本例ではＤ－ＲＡＭ２０２ａの所定領域）に１以上の液晶制御コマンドが記憶されているか否かを確認する。液晶制御コマンドがなければ、演出制御ＣＰＵ２００はステップＳ１０３０に進む。

液晶制御コマンドバッファＦ１５に１又は複数の液晶制御コマンドが記憶されていれば、演出制御ＣＰＵ２００はステップＳ１０２７の液晶制御コマンド解析処理として、液晶制御コマンドバッファＦ１５に記憶されているコマンドについての解析処理を行う。液晶制御コマンド解析処理においては、先の演出制御コマンド解析処理の場合と同様に、受信コマンドに応じて予め定められた処理を実行する。液晶制御コマンドのうちには、対応する処理として、液晶シナリオデータにシナリオ登録を行う処理が定められたものもあり、その場合には液晶シナリオデータに新たなシナリオが追加される。

例えばこのようにして、液晶制御コマンドの解析結果に基づいた液晶シナリオデータの生成が行われる。

【０３５６】

上述したステップＳ１００５の描画更新処理では、上記のように液晶制御コマンドの解析結果に基づいて生成された液晶シナリオデータに基づいてディスプレイリストの作成や、プリロードの実行開始制御が行われて、演出に必要な画像データが得られる。

なお、本例では、液晶表示装置３２としては主液晶表示装置３２Ｍと副液晶表示装置３２Ｓとが設けられているため、描画更新処理ではこれらの液晶表示装置に表示されるべき画像データについてのディスプレイリスト作成やプリロードがそれぞれ行われる。このように主液晶表示装置３２Ｍ、副液晶表示装置３２Ｓそれぞれについての処理を行う点は、ステップＳ１０５１のフレーム終了時処理における各処理（描画完了待ち、表示画面の切替え、プリロード完了待ち、描画開始）についても同様である。

【０３５７】

ステップＳ１０２７に続くステップＳ１０２８で演出制御ＣＰＵ２００は、先のステップＳ１０２４と同様にスケジューラ更新フラグをＯＦＦとし、ステップＳ１０３０に処理を進める。

【０３５８】

ここで、演出制御ＣＰＵ２００は、先のステップＳ１０２１、ステップＳ１０２５で処理時間が１６ｍｓ以下であった場合にはステップＳ１０３０に処理を進める。すなわち、演出制御コマンド、液晶制御コマンドの解析に係る処理は実行されない。

この場合の演出制御メイン処理では、フレーム期間の前半であれば、演出制御コマンド、液晶制御コマンドそれぞれの解析処理を必要なだけ何度でも行うことが許容されている。これにより、例えばフレーム期間の前半においてコマンドが連発した場合に、それらのコマンドを解析し、解析結果に応じたシナリオの更新（追加）を行うことが可能とされる。

【０３５９】
ステップＳ１０３０で演出制御ＣＰＵ２００は、スケジューラ更新フラグを確認して処理を分岐する。ここではスケジューラ更新フラグがＯＦＦの場合のみ、ステップＳ１０３１～Ｓ１０３４の処理を行うようにする。
スケジューラ更新フラグはフレーム開始直後のステップＳ１００９でＯＦＦとされるため、現在のフレーム期間で最初にステップＳ１０３０の処理に進んだときに、ステップＳ１３１～Ｓ１３４の処理を行うことになる。またステップＳ１０２４やＳ１０２８でスケジューラ更新フラグがＯＦＦとされるため、演出制御コマンドや液晶制御コマンドの受信があった場合もステップＳ１０３１～Ｓ１０３４の処理を行う。
【０３６０】
ステップＳ１０３１で演出制御ＣＰＵ２００は、スケジューラ更新フラグをＯＮとする。すなわち、以下のステップＳ１０３２～Ｓ１０３４の処理により各種スケジューラが更新されることに対応して予めスケジューラ更新フラグをＯＮとしておく。なお、各種スケジューラ更新が終了（本例ではＳ１０３４が終了）してからスケジューラ更新フラグをＯＮとしてもよい。
【０３６１】
ステップＳ１０３２で演出制御ＣＰＵ２００は、シナリオスケジューラ実行としての処理を行う。これはシナリオ登録情報（図２０Ａ参照）の更新処理であり、具体的には、図１８に示したシナリオ更新処理と同様の処理を行う。例えば、シナリオチャネルｓＣＨに登録された情報のうち待機時間（delay）が０のものについて、登録されたデータに対応した処理を実行する。即ちメインシナリオテーブルのうちで指定されたシナリオ番号に応じた処理を行う。
ここで、前述のように第二実施形態では、シナリオ登録情報における待機時間（delay）、メインシナリオタイマ（mcTm）の更新はステップＳ１００５のシナリオスケジューラフレーム更新処理にて１フレームごとに行われる。すなわち、この場合に実行する図１８のシナリオ更新処理においては、ステップＳ６０２の待機時間更新処理、及びステップＳ６０９のタイマ更新処理は省略される。
【０３６２】
ステップＳ１０３３で演出制御ＣＰＵ２００は、サウンドスケジューラ実行及び出力の処理を行う。これは、シナリオテーブルで指定される音サブシナリオに応じた音データ登録情報（図２１参照）の更新処理、及び音データ登録情報に基づいた音制御信号の出力処理である。演出制御ＣＰＵ２００は、音コントローラ２３０に対して、現在のシナリオ進行に応じた音出力を実行させる。
また、ステップＳ１０３４で演出制御ＣＰＵ２００は、ランプスケジューラ実行処理を行う。これはシナリオテーブルで指定されるランプサブシナリオに応じたランプデータ登録情報（図２０Ｂ参照）の更新処理である。
【０３６３】
ステップＳ１０３３、Ｓ１０３４の処理としては、図１９に示したサブシナリオの更新処理と同様の処理を行う。即ち、ステップＳ１０３３におけるサウンドスケジューラ実行に係る処理としては、図１９におけるステップＳ６２１の処理とステップＳ６３１以降の処理と同様の処理を実行する。但し、第二実施形態では、先のステップＳ１００８のサウンドスケジューラフレーム更新処理によってサブシナリオタイマ（scTm）の値の更新が行われるため、この場合に実行するステップＳ６２１及びＳ６３１以降の処理においては、ステップＳ６４０のタイマ更新処理、及び該タイマ更新処理を実行するか否かの条件判定を行うステップＳ６３９の判別処理は省略される。
ステップＳ１０３４のランプスケジューラ実行処理としては、図１９のステップＳ６２１～Ｓ６３０と同様の処理を実行する。この処理により、ランプデータ登録情報は、実行すべきランプ演出動作が登録された状態に更新される。
なお、このように更新されるランプデータ登録情報に基づくランプ駆動データ（前述した「ＬＥＤ駆動データ」と同義）の作成及びランプ駆動データ出力は、後のステップＳ１

０４１、Ｓ１４０２で行われる。
【０３６４】
ここで、先の図４１の説明時に言及したが、本例では、演出進行を管理するためのシナリオデータとしては、ランプ、音、モータ以外にも液晶制御コマンドの送信管理を行うためのコマンド送信シナリオデータも存在する。
図４３での図示は省略したが、該コマンド送信シナリオデータについてのスケジューラ実行処理（登録されたメインシナリオ番号に応じたコマンド送信シナリオデータに対するデータ登録、及び登録データが表す液晶制御コマンドの液晶制御コマンドバッファＦ１５へのセット処理）は、例えば、ステップＳ１０３２のシナリオスケジューラ実行処理の一部として実行することが考えられる。或いは、例えばステップＳ１０３４のランプスケジューラ実行処理のように、シナリオスケジューラ実行処理とは別処理として実行することもできる。
【０３６５】
上記ステップＳ１０３２～Ｓ１０３４による各種スケジューラに係る処理を実行したことに応じ、演出制御ＣＰＵ２００はステップＳ１０４０で、処理時間が２０ｍｓ未満であるか否かにより処理を分岐する。処理時間が２０ｍｓ未満であればステップＳ１０４１～Ｓ１０４３の処理を実行し、２０ｍｓ未満でなければこれらの処理を実行しない。
ここで、２０ｍｓの意図は、フレームの前半期間を多少経過しても、以下のステップＳ１０４１、Ｓ１０４２によるランプ駆動データの更新や出力の処理を実行可能とすることにある。つまり２０ｍｓはあくまで一例であり、該意図に沿った範囲内であれば値の変更が可能である。
【０３６６】
ステップＳ１０４１で演出制御ＣＰＵ２００は、ランプ駆動データ更新処理を行う。即ち、ランプデータ登録情報の各ランプチャネルｄｗＣＨに登録されている情報に基づいて、実際にシリアル出力コントローラ２４０から枠ドライバ部６１及び盤ドライバ部６２に出力すべきランプ駆動データ（ＬＥＤ駆動データ）を生成する処理である。このステップＳ１０４１の処理としては、図３６に示したＬＥＤ駆動データ更新処理と同様の処理を行う。但し、第二実施形態では、先のステップＳ１００７のランプスケジューラフレーム更新処理によりランプデータ登録情報（図２０Ｂ）における実行時間（time）の値が更新されるため、この場合において実行する図３６の処理では、ステップＳ７１０の更新処理は省略される。
【０３６７】
そしてステップＳ１０４２で演出制御ＣＰＵ２００は、ランプ駆動データ出力処理として、生成したランプ駆動データをシリアル出力コントローラ２４０に転送し、シリアルデータとして出力させるように制御する。ステップＳ１０４２の出力処理としては、例えば、図２５に示したＬＥＤ駆動データ出力処理と同様の処理を行う。
【０３６８】
さらに、続くステップＳ１０４３で演出制御ＣＰＵ２００は、キーイベント処理を行い、ステップＳ１００２に戻る。
【０３６９】
現在のフレーム期間において、処理時間が３１ｍｓ以上となった場合、演出制御ＣＰＵ２００はステップＳ１０２０からＳ１０５０に処理を進めて、Ｖブランク信号が発生するまで待機する。なお、ここでのＶブランク信号は、フレーム期間開始から３１ｍｓ経過後のＶブランク信号（２回目のＶブランク信号）であるので、ステップＳ１０５０の処理はフレーム期間の終了を待機する処理と換言できる。
【０３７０】
ステップＳ１０５０でＶブランク信号が検知されると、演出制御ＣＰＵ２００はＳ１０５１のフレーム終了時処理を行う。該終了時処理では、描画完了待ちの処理、表示画面の切替え、プリロード完了待ち、描画開始等の処理が行われる。
【０３７１】

フレーム終了時処理の具体例を図４４に示す。
先ず、ステップＳ１０７１で演出制御ＣＰＵ２００は描画の完了を待機する。これは次のフレーム期間に表示する画像の描画完了を待機するものである。
前述のようにＶＲＡＭ２０９にはフレームバッファ２０９Ａ、２０９Ｂが用意され、表示回路２２１等が一方の表示データを読み出して表示させているフレーム期間には、他方のフレームバッファに、次のフレームの表示データの描画が行われる。この描画が完了していることをステップＳ１０７１で確認する。
【０３７２】
なお通常は、このフレーム期間の終了時点では、次のフレームの表示データの描画は完了しているように設計されている。この時点で描画が完了していないことは、描画処理に何らかの不具合がある場合と想定される。
そこで、描画完了待機が発生した場合、その待機時間をカウントして、待機時間によっては、１フレーム期間にアクセスされる描画数を減らすなどの変更を行うようにしてもよい。
また、多少（１ｍｓ程度）の待機であれば、次のフレームの処理が少し遅れるだけであるため、特に問題はないというようにしてもよい。
なお、いつまでも描画が終わらない場合、演出制御ＣＰＵ２００の処理についてＷＤＴ回路２１０によるリセットがかかる。
【０３７３】
描画完了を確認したら演出制御ＣＰＵ２００はステップＳ１０７２で、表示画面のスワップを行う。即ちＶＲＡＭ２０９のフレームバッファ２０９Ａ、２０９Ｂについて、表示データを読み出すフレームバッファの切替を行う。
なお、ステップＳ１０５０でフレーム期間終了が検知されたことで無条件にフレームバッファ切替を行うことも考えられるが、本実施形態の場合、ステップＳ１０７１で描画完了確認を行ってからフレームバッファ切替が行われる。このため、表示回路２２１等に転送される表示データは、正しく描画が完了した表示データとなる。つまり、描画が垂直同期信号の２回目の時点までに終わらなかったにも拘わらず描画中のフレームバッファの表示データが表示されてしまうといったことは生じない。
【０３７４】
ステップＳ１０７３では演出制御ＣＰＵ２００は、プリロードの完了を確認する。
通常は、プリロードは１フレーム期間内に終了するように設計されているが、このステップＳ１０７３で確認している。
なお、多少プリロードが遅れたとしても、上記のステップＳ１０７２で既に次のフレームの表示データの出力は開始されるため、大きな問題とはならない。
その後ステップＳ１０７４で、プリロードされたディスプレイリストを先のステップＳ１０７２で表示データ読み出し対象に切り替えられたフレームバッファとは逆のフレームバッファに書き込む処理を行う。
【０３７５】
以上のフレーム終了時処理を行ったら、図４３のステップＳ１０５２においてフレーム更新フラグをＯＦＦとし、ステップＳ１００２に戻る。そして説明してきたように、新たなフレーム期間についての処理を行う。
【０３７６】
以上の図４３の処理とともに、演出制御ＣＰＵ２００では例えば１ｍｓ毎の割込み処理として、図４５の処理が実行される。即ち図４５の処理は、表示データのフレームタイミングに関わらず１ｍｓ毎に実行される処理である。
演出制御ＣＰＵ２００はステップＳ１１００で、ＷＤＴパルス生成処理を行う。ＷＤＴ回路２１０はこのＷＤＴパルスにより１ｍｓ毎のカウントを行う。
ステップＳ１１０１で演出制御ＣＰＵ２００は、モータ、ソレノイドのセンサ更新処理を行う。これは上述した原点スイッチ６８の情報を検知する処理である。
【０３７７】

ステップＳ１１０２で演出制御ＣＰＵ２００はデバイススケジューラ実行処理を行う。具体的にはモータデータ登録情報（図２０Ｃ参照）の更新を行う。またステップＳ１１０３でデバイススケジューラ出力処理として、モータドライバ７０へのモータ駆動データの出力処理を行う。
即ち演出制御ＣＰＵ２００は、フレーム期間に比べて約１／３０の時間間隔である１ｍｓ毎に、モータ動作制御を行っている。
ここで、ステップＳ１１０２のデバイススケジューラ実行処理、ステップＳ１１０３のデバイスススケジューラ出力処理は、先の図１２に示したステップＳ２０３のモータ動作更新処理、ステップＳ２０４のモータ出力処理と同様となるため重複説明は避ける。
【０３７８】
続くステップＳ１１０４で演出制御ＣＰＵ２００は、キー入力処理を行う。即ち、操作部６０における各種操作子（演出ボタン１１，１２、十字キー１３等）の操作に応じた信号を確認する。
なお、図３９に示したように、原点スイッチ６８の検出情報と操作部６０の操作情報は、シリアル入力データとして検知することができる。そこで、キー入力処理は、ステップＳ１１０１で原点スイッチ６８の検出情報と同時に確認するようにしてもよい。これにより検出処理を効率化できる。
【０３７９】
ここまで、演出制御部５１Ａにおける演出制御ＣＰＵ２００の処理例を説明してきたが、以上の処理例では、演出制御ＣＰＵ２００は表示データのフレーム周期に合わせて各種処理を行っている。
図４６は、表示データのフレーム期間内における各種タイミングを示している。
描画回路２２５による描画処理は、フレーム期間内に、次のフレームの表示データについて実行される。例えばフレームＦＲ１の表示データの描画は、フレームＦＲ０の表示期間に実行される。具体的にはステップＳ１０５１で描画が開始されるため、フレーム先頭時点で、次のフレームのための描画が開始される。この描画は、通常はフレーム期間内に完了する。
もし、フレーム期間内に描画が完了していないとすると、図４４のステップＳ１０７１で完了が待機されることになる。
【０３８０】
プリローダ２２０によるプリロードは、描画のための準備として、さらに１フレーム前の期間に実行される。例えばフレームＦＲ２の表示データの描画のためのプリロードは、フレームＦＲ０のフレーム期間内に実行される。具体的にはステップＳ１００５で開始されるため、フレーム先頭時点で、２つ後のフレームのためのプリロードが開始される。プリロードとしても、通常はフレーム期間内に完了する。
もし、プリロードがフレーム期間内に完了していないとすると、図４４のステップＳ１０７３で完了が待機されることになる。プリロードが多少遅れると、それによって描画の開始が多少遅れることになる。
【０３８１】
演出制御ＣＰＵ２００の処理としては、各フレームの開始タイミングにおいて、ステップＳ１０５１の処理が行われることになる。主な処理としては、図中に示すように、フレームバッファ２０９Ａ、２０９Ｂの切替（ＦＢａ＜＝＞ＦＢｂ）、及び描画開始が先ずは行われる。
次いで、ステップＳ１００４の液晶制御フレーム更新（液晶シナリオの更新）、ステップＳ１００５によるディスプレイリスト作成とプリロード実行開始、及びステップＳ１００６～Ｓ１００８による各種スケジューラフレーム更新（シナリオ、ランプ、サウンド）が行われる。
そして、これらステップＳ１００４～Ｓ１００８の処理が１度行われた後は、必要に応じてステップＳ１０２３の演出制御コマンド解析処理、ステップＳ１０２６の液晶制御コマンド解析処理、及びステップＳ１０３２～Ｓ１０３４の各種スケジューラ実行処理（シ

ナリオ、サウンド、ランプ、及び本例ではコマンド送信シナリオのスケジューラ更新を含む）が行われる。

さらに、ステップＳ１０４１、Ｓ１０４２によるランプ駆動データの更新、出力の各処理と、ステップＳ１０４３のキーイベント処理が実行される。

【０３８２】

そして、以上までの処理がフレーム期間の開始時から１６ｍｓ以内に完了していれば、再度、演出制御コマンドや液晶制御コマンドについての解析処理を行うことが可能とされ、解析処理が行われればさらに各種スケジューラ実行処理、及びランプ駆動データ更新・出力処理、及びキーイベント処理が行われる。

このとき、ランプ駆動データ更新・出力処理及びキーイベント処理については、フレーム期間の開始時から２０ｍｓ以内であれば実行可能とされている（Ｓ１０４０参照）。すなわち、直前の各種スケジューラ実行処理の完了タイミングがフレーム期間の後半に多少及んだとしても、これらの処理を実行することができる。

ランプ駆動データ更新・出力処理について、フレーム期間開始時からの経過時間に応じて処理の実行を不許可とすることで、これらランプ駆動データの更新や出力の処理によってフレーム終了タイミングに対してフレーム終了処理（Ｓ１０５１）が遅れることを防止している。

なお、仮にランプ駆動データ更新・出力処理が不許可とされた場合には、シリアル出力コントローラ２４０からのランプ駆動データの出力が行われないことが想定されるが、その場合、ＬＥＤドライバ９０は直前のランプ駆動データを維持しているため、直前の発光動作が実行されることになる。

【０３８３】

［2-5. 演出制御の動作例］

図４７～図４９を参照し、上記した演出制御部５１Ａ（演出制御ＣＰＵ２００）の処理により実現される演出制御の動作例を説明する。

以下の動作例では、大当たりの終了時に対応して行われる演出制御の動作例を説明する。

【０３８４】

図４７Ａは、主制御部５０が送信する演出制御コマンドの種類の例を示している。

演出制御コマンドとしては、上位１バイトの値によって図のように種々のコマンドが定義されている。そして、各コマンドには、対応して実行すべき処理（対応処理）が予め定められている（本例では「関数」として定められている）。

本例で着目するのは、大当たり終了指定コマンドとしての演出制御コマンド（「０ｘＦ３」）である。

【０３８５】

図４７Ｂは、大当たり終了指定コマンドの対応処理において送信が定められた液晶制御コマンドの一覧を例示している。

大当たり終了指定コマンドに対しては、当該大当たり終了指定コマンド（Ｆ３０１Ｈ）のスルー送信が定められていると共に、選択キャラコマンド（０２ｘｘＨ）、連荘数コマンド（０３ｘｘＨ）、攻略フラグコマンド（０５ｘｘＨ）、大当たり図柄コマンド（０ＡｘｘＨ）、大当たり根幹コマンド（８８ｘｘＨ）、決定時のキャラコマンド（９ＦｘｘＨ）、及び背景指定コマンド（０１ｘｘＨ）の送信が定められている。

【０３８６】

選択キャラコマンドは、遊技者が選択したキャラクタ（例えば好みのキャラクタ）の別を表すコマンドであり、連荘数コマンドは大当たりの連続回数を表すコマンドである。攻略フラグコマンドは、選択したキャラクタを遊技者が攻略できたか否かを表すコマンド、大当たり図柄コマンドは今回の当たり目としての図柄を表すコマンド、大当たり根幹コマンドは今回の大当たりの種類、例えばビッグやスモールの大当たりや何ラウンドの大当たりか等を表すコマンドである。

【０３８７】
　決定時のキャラコマンド、及び背景指定コマンドは、大当たり終了指定コマンドの受信時から所定時間後の送信（つまり前述した時間差送信）が指定されたコマンドである。本例のパチンコ遊技機１Ａでは、大当たり終了に応じて遊技者にキャラクタを選択させる。例えば、主液晶表示装置３２Ｍ等に表示したキャラクタのうちから特定のキャラクタを選択させるものであり、演出制御５１Ａは、演出ボタン１２等の所定操作子に対する操作を遊技者によるキャラクタ選択についての決定操作として受け付ける。
　決定時のキャラコマンドは、このような遊技者の操作により決定されたキャラクタの種別を表すコマンドとされる。
　背景指定コマンドは、例えば主液晶表示装置３２Ｍの表示画像における背景画像の種別を指定するコマンドとされる（例えば、決定されたキャラクタに応じた背景画像が指定される）。
　本例では、決定時のキャラコマンド及び背景指定コマンドは、大当たり終了指定コマンドの受信から略９ｓ（２７０フレーム）の時間差を以て送信すべきコマンドとして定められている。換言すれば、この場合における大当たり終了時の演出では、大当たり終了から９ｓまでの間に遊技者によるキャラクタの選択（決定）操作が受け付けられ、その後、決定されたキャラクタに応じた決定時のキャラコマンド及び背景指定コマンドの送信（液晶制御コマンドバッファＦ１５へのセット）が行われる。
【０３８８】
　本例では、時間差送信が定められた決定時のキャラコマンド、及び背景指定コマンドについては、その送信管理のために前述したコマンド送信シナリオデータが用いられる。
　具体的に、本例の演出制御部５１Ａは、受信した演出制御コマンドが大当たり終了指定コマンドであった場合のコマンド解析処理（ステップＳ１０２３）において、時間差送信の対象外である大当たり終了指定コマンド（スルー送信）、選択キャラコマンド、連荘数コマンド、攻略フラグコマンド、大当たり図柄コマンド、及び大当たり根幹コマンドについては、液晶制御コマンドバッファＦ１５にセットする処理が行われる。
　そして、この場合のコマンド解析処理（Ｓ１０２３）では、受信した大当たり終了指定コマンドに対応して定められたメインシナリオ番号（mcNo）をシナリオ登録情報に登録する処理が行われるが、このように登録されるメインシナリオにおいて、大当たり終了時に対応したランプや音、モータのシナリオが定められていると共に、９ｓ後に決定時のキャラコマンド及び背景指定コマンドを送信するシナリオが定められている。つまり、上記のように大当たり終了指定コマンドに対応したメインシナリオ番号（mcNo）の登録が行われることで、その後の各種スケジューラ実行処理（Ｓ１０３２～Ｓ１０３４）により大当たり終了時に対応したランプ、音、モータの各演出についてのシナリオ（サブシナリオ）の登録、及びコマンド送信シナリオの登録が行われる。なお、前述のように、コマンド送信シナリオデータについてのスケジューラ実行処理は、例えば、ステップＳ１０３２のシナリオスケジューラ実行処理の一部として実行することができる。
【０３８９】
　図４８は、大当たり終了指定コマンドに対応して定められた音、ランプ、モータ、ボタン、コマンド送信についての各シナリオデータを模式的に表している。
　なお、図中の時間（ｆ）は演出の進行時間を表すもので、単位は「フレーム」（表示データのフレーム）である。この場合の大当たり終了時の演出シナリオは、３００フレーム分の時間長に及ぶものとされている。時間（ｆ）について、括弧内の数値は対応するラインのシナリオ実行時間を表している。これは、前述したランプや音／モータについての実行時間（time）に相当する（図２３Ａ、図２３Ｂ参照）。
【０３９０】
　図示するようにこの場合のシナリオデータにおいては、時間＝０のラインにおいて音、ランプ、モータについてのサブシナリオデータをそれぞれセットし、さらに時間＝１２０のラインにおいて、ランプのサブシナリオデータとしてボタンランプの点灯を表すデータをセットすることが定められている。ここでのボタンは、前述したキャラクタの決定操作

に用いられる操作子を意味するものであり、ボタンランプとは該操作子を発光させるためのランプを意味する。ここでのボタンは、本例では演出ボタン１２であるとする。

また、時間＝１２３のライン、時間＝１４４のラインにおいて、ボタンの有効開始、有効終了を表す情報をセットすることが定められている。ここでの有効／無効は、操作受け付けの有効／無効を意味する。

さらに、時間＝２７０のラインでは、ランプのサブシナリオデータとしてボタンランプの消灯を表すデータをセットし、またコマンド送信シナリオデータとして決定時のキャラコマンド、及び背景指定コマンドの送信を表すデータをセットすることが定められている。

時間＝３００のラインには、シナリオ終了を表すコードが記述される（前述した終了コードD_LSENDやD_SEENDを参照）。

【０３９１】

なお、ここでは、ボタンの有効／無効の管理をシナリオデータに基づき行う例を示したが、ボタンの有効／無効を管理する手法としては必ずしもシナリオデータを用いた手法に限定されない。

【０３９２】

図４９は、大当たり終了指定コマンドの受信に応じて演出制御部５１Ａが実行する処理の流れを説明するための図である。

図４９において、横軸は1フレーム期間内での経過時間を表し、縦軸はフレーム単位での経過時間を表している。参考として、最上段には、１フレーム期間内における図４３の各処理の流れを図４６と同様に示している。

【０３９３】

とあるｎフレーム目において、主制御部５０からの大当たり終了指定コマンドが受信されたとする。コマンドの受信タイミングは、図のように各種スケジューラフレーム更新処理（Ｓ１００６～Ｓ１００８）を実行中のタイミングであったとする。なお、コマンドの受信処理は、前述のように外部割込み処理において実行される。

【０３９４】

この場合、当該ｎフレーム目の最初の演出制御コマンド解析処理（Ｓ１０２３）において、受信した大当たり終了指定コマンドに対応した処理が実行される。すなわち、上述した大当たり終了指定コマンド（スルー送信）、選択キャラコマンド、連荘数コマンド、攻略フラグコマンド、大当たり図柄コマンド、及び大当たり根幹コマンドを液晶制御コマンドバッファＦ１５にセットする処理（液晶制御へのコマンド送信処理）と、大当たり終了指定コマンドに対応して定められたメインシナリオ番号（mcNo）のシナリオ登録情報への登録が行われる。

【０３９５】

そして、続く液晶制御コマンド解析処理（Ｓ１０２７）において、液晶制御コマンドバッファＦ１５にセットされた大当たり終了指定コマンド、選択キャラコマンド、連荘数コマンド、攻略フラグコマンド、大当たり図柄コマンド、及び大当たり根幹コマンドについての解析処理が行われる。

【０３９６】

先に説明したように、液晶制御コマンド解析処理によっては、液晶シナリオデータにシナリオ登録が行われる。そして、ステップＳ１００５の描画更新処理において、液晶シナリオデータに基づいたディスプレイリストの作成や、プリロードの実行開始制御が行われて、演出に必要な画像データが得られる。

ｎフレーム目で更新された液晶シナリオデータに基づく液晶シナリオは、ｎ＋１フレーム目の液晶制御フレーム更新処理（Ｓ１００４）によってシナリオ進行タイミングを管理する値の更新が行われることで開始され、その直後の描画更新処理（Ｓ１００５）によって該開始されたシナリオに応じた描画更新のための処理が行われる。

【０３９７】

ｎフレーム目において、上記した液晶制御コマンド解析処理の直後に実行される各種ス

ケジューラ実行処理（Ｓ１０３２～Ｓ１０３４）によっては、登録されたメインシナリオ番号に対応したランプ、音についてのサブシナリオが登録され開始される。さらに、直後のランプ駆動データ更新・出力処理（Ｓ１０４１、Ｓ１０４２）により、登録されたランプサブシナリオに応じたランプ制御を行うための駆動データ（ＬＥＤ駆動データ）の更新・出力が開始される。すなわち、対応するランプ演出が開始される。

【０３９８】

また、受信した演出制御コマンドが大当たり終了指定コマンドであった場合には、コマンド送信シナリオとして決定時のキャラコマンド及び背景指定コマンドの時間差送信（２７０フレーム後）に係るシナリオが登録される。

このため、ｎ＋２７０フレーム目における各種スケジューラ実行処理によっては、コマンド送信シナリオデータに従って、これら決定時のキャラコマンド及び背景指定コマンドとしての液晶制御コマンドの送信が行われる。また、時間＝２７０のラインには、ボタンランプの消灯に係るサブシナリオが登録されており、この場合の各種スケジューラ実行処理によっては当該消灯に係るサブシナリオが開始され、その直後に行われるランプ駆動データ更新・出力によって対応するランプ駆動データの出力が開始される。

ｎ＋２７０フレーム目において、上記の各種スケジューラ実行処理によって送信された決定時のキャラコマンド及び背景指定コマンドについては、当該ｎ＋２７０フレーム目に実行される液晶制御コマンド解析処理において解析が行われ得る。図４９の例では、ランプ駆動データ出力処理がフレーム期間開始時から１６ｍｓよりも十分に早いタイミングで完了したことにより、ｎ＋２７０フレーム目における液晶制御コマンド解析処理によってこれらコマンドの解析が行われた例を示している。前述のように、液晶制御コマンド解析処理は、フレーム期間開始時から１６ｍｓ以内であれば実行され得るものである。

【０３９９】

図示は省略したが、決定時のキャラコマンド及び背景指定コマンドについても、上記した時間差送信の対象外の各種コマンド（選択キャラコマンド等）と同様にコマンド解析結果に基づく液晶シナリオデータへの新たなシナリオの登録が行われ、該新たなシナリオは、次フレームにおける液晶制御フレーム更新処理によって開始され、その直後の描画更新処理によって該開始されたシナリオに応じた描画更新のための処理が行われる。

【０４００】

ここで、演出制御コマンド、液晶制御コマンドの解析処理は、比較的処理負担の重い処理となる共に、受信コマンド数によって処理も多くなる。コマンド種類によっては演出のための抽選等も行う必要が生ずる。

本実施形態では、このような処理をフレーム前半期間にのみ行うものとしている。

これにより、以降に続く各種スケジューラ実行処理やランプ駆動データ更新・出力処理がフレーム期間内に終了しなくなる事態の発生防止を図っている。

【０４０１】

また、一方で本実施形態では、フレーム前半期間であれば演出制御コマンド、液晶制御コマンドの解析処理を複数回行うことを許容している。これにより、限られた処理期間において効率的に受信コマンドの解析を行うことができる。

【０４０２】

また、本実施形態では、ランプ駆動データの更新・出力処理についてはフレーム前半期間を多少過ぎても実行を許可するものとしている。これにより、ランプ駆動データの更新・出力処理が実行されないことに伴いランプ駆動データとして直前フレームのデータ（つまり未更新のままのデータ）が維持されてしまう事態の発生回数を低減でき、ランプ演出の正確性向上を図ることができる。

【０４０３】

［2-6．第二実施形態のまとめ及び変形例］

以上の第二実施形態のパチンコ遊技機１Ａによれば、演出制御のための処理を効率化でき、適正な演出制御を実現できる。

また演出制御処理負荷を軽減することができる。

【０４０４】

また、第二実施形態の遊技機（パチンコ遊技機１Ａ）は、第一の構成として、遊技動作の進行制御を行う主制御手段（主制御部５０）と、主制御手段からの指示情報（演出制御コマンド）に基づき演出制御を行う演出制御手段（演出制御部５１Ａ）と、を備え、演出制御手段は、画像表示以外の演出の実行を管理する第一手段（演出制御シナリオ実行処理部Ｆ１３）と、画像表示に関する演出の実行を管理する第二手段（液晶制御コマンド解析処理部Ｆ１６及び液晶制御シナリオ実行処理部Ｆ１７）と、を有し、主制御手段からの指示情報を記憶する指示情報記憶手段（演出制御コマンドバッファＦ１１）と、指示情報記憶手段に記憶された指示情報を解析する解析手段（演出制御コマンド解析処理部Ｆ１２）と、解析手段の解析結果によって得られた第二手段についての指示情報である第二指示情報（液晶制御コマンド）を記憶する第二指示情報記憶手段（液晶制御コマンドバッファＦ１５）と、をさらに有している。

【０４０５】

上記のように演出制御手段が第二指示情報記憶手段を有していることで、当該演出制御手段をいわゆる１ＣＰＵとして構成する場合には、主制御手段からの指示情報を解析し、解析により得た第二指示情報を所定記憶領域にセットするという処理については、２ＣＰＵの場合における演出制御手段（演出制御部５１）の処理を踏襲することが可能とされる。

また、画像演出の制御処理についても、第二指示情報記憶手段にセットされた第二指示情報を取得し、該第二指示情報に基づいて画像表示装置を制御するという、２ＣＰＵの場合における画像演出制御手段（液晶制御基板５２の液晶制御ＣＰＵ）と同様の処理を踏襲することが可能とされる。

このように、第二指示情報記憶手段を有していることで、１ＣＰＵとしての演出制御手段が実行すべきプログラムとしては、２ＣＰＵ時の演出制御手段が実行していたプログラムと画像演出制御手段が実行していたプログラムをほぼそのまま踏襲することが可能とされる。

従って、１ＣＰＵを実現するにあたって既存プログラムの書き替えをほぼ不要とすることができ、プログラム作成の作業負担軽減が図られ、コスト削減を図ることができる。

また、画像演出制御についてのプログラムは、１ＣＰＵの場合と２ＣＰＵの場合とでほぼ共通なものとすることができ、従ってプログラムの汎用性を高めることができる。同様に、画像演出制御以外の制御についてのプログラムとしても、１ＣＰＵの場合と２ＣＰＵの場合とでほぼ共通なものとすることができ、汎用性を高めることができる。

【０４０６】

さらに、上記した第一の構成としての遊技機においては、第二指示情報がコマンド情報とされている。

これにより、既存の２ＣＰＵ構成における第二指示情報がコマンド情報とされている場合に対応して、既存プログラムからの変更点をさらに少なくすることが可能とされる。

従って、１ＣＰＵ化を実現するにあたってのプログラム作成の作業負担をさらに軽減でき、さらなるコスト削減を図ることができる。また、プログラムの汎用性をさらに高めることができる。

【０４０７】

また、第二実施形態の遊技機は、第二の構成として、遊技動作の進行制御を行う主制御手段（主制御部５０）と、主制御手段からの指示情報（演出制御コマンド）に基づき演出制御を行う演出制御手段（演出制御部５１Ａ）と、を備え、演出制御手段は、画像表示以外の演出の実行を管理する第一手段（演出制御シナリオ実行処理部Ｆ１３）と、画像表示に関する演出の実行を管理する第二手段（液晶制御コマンド解析処理部Ｆ１６及び液晶制御シナリオ実行処理部Ｆ１７）と、主制御手段からの指示情報を解析する解析手段（演出制御コマンド解析処理部Ｆ１２）と、を有し、第一手段は、解析手段による解析結果に基づいて、画像表示以外の演出の進行を管理する第一演出進行管理手段（演出制御シナリオ

実行処理部Ｆ１３）と、第一演出進行管理手段による演出の進行に応じて、画像表示に関する画像表示指示情報（液晶制御コマンド）を生成する画像表示指示情報生成手段（演出制御シナリオ実行処理部Ｆ１３）と、を有し、第二手段は、画像表示指示情報生成手段により生成された画像表示指示情報に基づいて、画像表示に関する演出の進行を管理する第二演出進行管理手段（液晶制御コマンド解析処理部Ｆ１６及び液晶制御シナリオ実行処理部Ｆ１７）と、を有している。

【０４０８】

２ＣＰＵ構成の場合、画像演出制御手段（液晶制御基板５２の液晶制御ＣＰＵ）は、演出制御手段（演出制御部５１）が主制御手段からの指示情報を解析して得た画像表示指示情報を取得し、該画像表示指示情報に基づいて画像表示装置の制御を行って所要の画像演出動作を実現させている。

このため、上記のような第二手段（第二演出進行管理手段）を有していることで、画像演出に係る制御については、画像演出制御手段のプログラムをほぼそのまま踏襲することが可能とされる。

また、第一手段による処理は、２ＣＰＵ構成における演出制御手段が実行していた処理とほぼ同様であるため、該処理についても２ＣＰＵ時に用いていたプログラムをほぼそのまま踏襲することが可能とされる。

このように１ＣＰＵとしての演出制御手段が実行すべきプログラムとしては、２ＣＰＵ時の演出制御手段が実行していたプログラムと画像演出制御手段が実行していたプログラムをほぼそのまま踏襲することが可能とされる。

従って、１ＣＰＵを実現するにあたって既存プログラムの書き替えをほぼ不要とすることができ、コスト削減を図ることができる。

また、第一手段、第二手段をそれぞれ実現するためのプログラムは２ＣＰＵの場合とほぼ共通なものとすることができ、従ってプログラムの汎用性を高めることができる。

【０４０９】

また、上記した第二の構成としての遊技機においては、画像表示指示情報がコマンド情報とされている。

【０４１０】

これにより、既存の２ＣＰＵ構成における画像表示指示情報がコマンド情報とされている場合に対応して、既存プログラムからの変更点をさらに少なくすることが可能とされる。

従って、１ＣＰＵ化を実現するにあたってのプログラム作成の作業負担をさらに軽減でき、さらなるコスト削減を図ることができる。また、プログラムの汎用性をさらに高めることができる。

【０４１１】

なお、本発明は実施形態で挙げた例に限らず多様な変形例や適用例が考えられる。

上記では、表示データのフレームに同期する信号としてＶブランク信号に基づいてタイミングを管理する例を挙げたが、Ｖブランク信号以外の垂直同期信号を用いたり、或いは水平同期信号のカウントでフレーム期間内のタイミング管理を行っても良い。

【０４１２】

また、上記では、画像表示手段として液晶表示装置を用いる例を挙げたが、例えば有機ＥＬ（Electro Luminescence）表示装置等の他の画像表示手段を用いることもできる。

【０４１３】

また、上記では、第二実施形態のような１ＣＰＵの構成において、第一実施形態で説明したランプの発光駆動情報格納手法を適用する例を挙げたが、第二実施形態において、発光駆動情報格納手法は例えば先行例の場合と同様とする等、第一実施形態の手法に限定されない。

【０４１４】

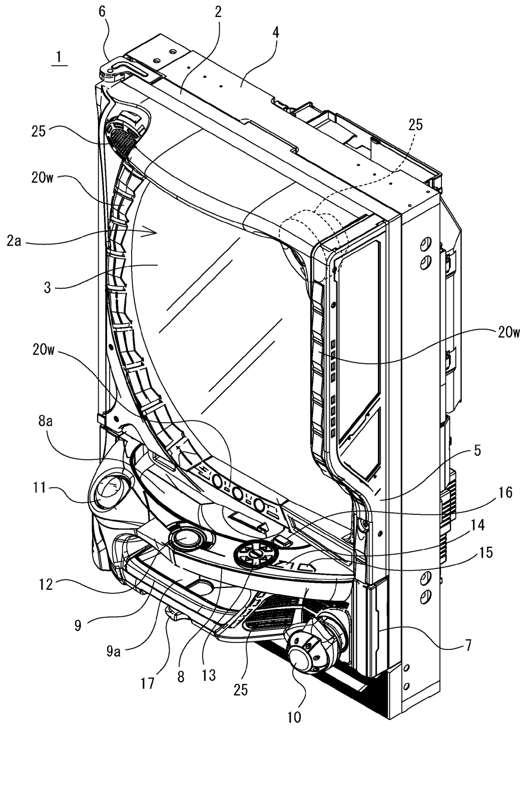
また、上記では、本発明をパチンコ遊技機１又は１Ａのような弾球遊技機に適用する例を示したが、回胴式遊技機（いわゆるスロット機）にも適用することができる。

【符号の説明】

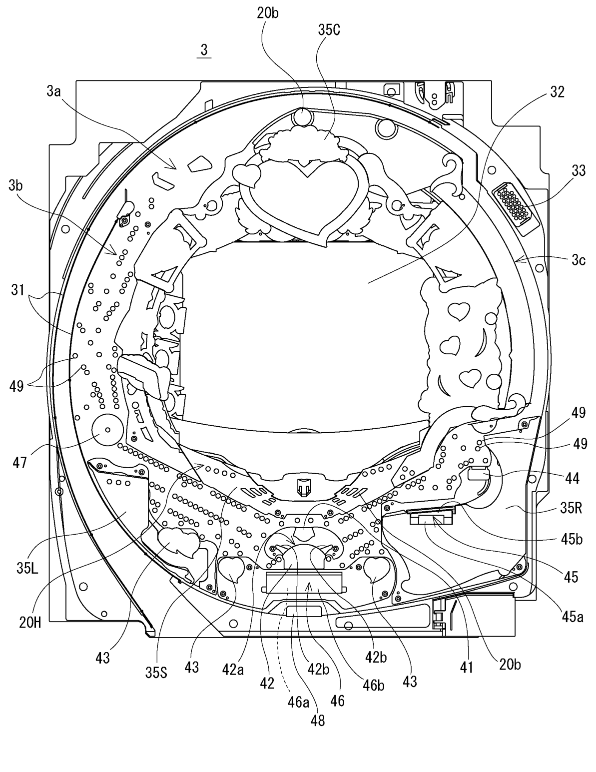
【０４１５】

１　パチンコ遊技機
２０ｗ，２０ｂ　発光部
３２　液晶表示装置
３３　図柄表示部
４１　上始動口
４２　普通変動入賞装置
４２ａ　下始動口
４３　一般入賞口
４４　ゲート
４５　第１特別変動入賞装置
４６　第２特別変動入賞装置
５０　主制御基板
５１　演出制御基板
５２　液晶制御基板
６１　枠ドライバ部
６２　盤ドライバ部
６３，６４　ランプ部
９０　ＬＥＤドライバ
１２０　ＬＥＤ
２００　演出制御ＣＰＵ
２０１　演出制御ＲＯＭ
２０２　演出制御ＲＡＭ

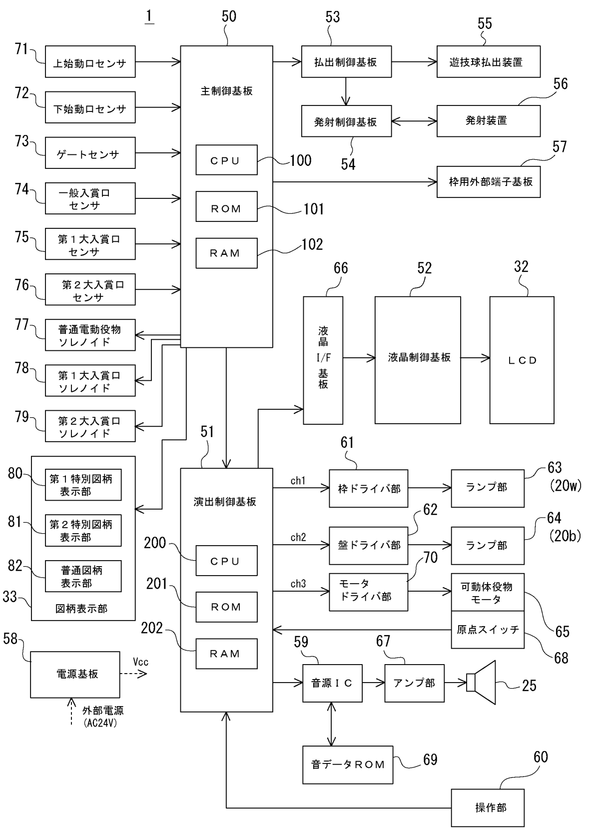
【図１】



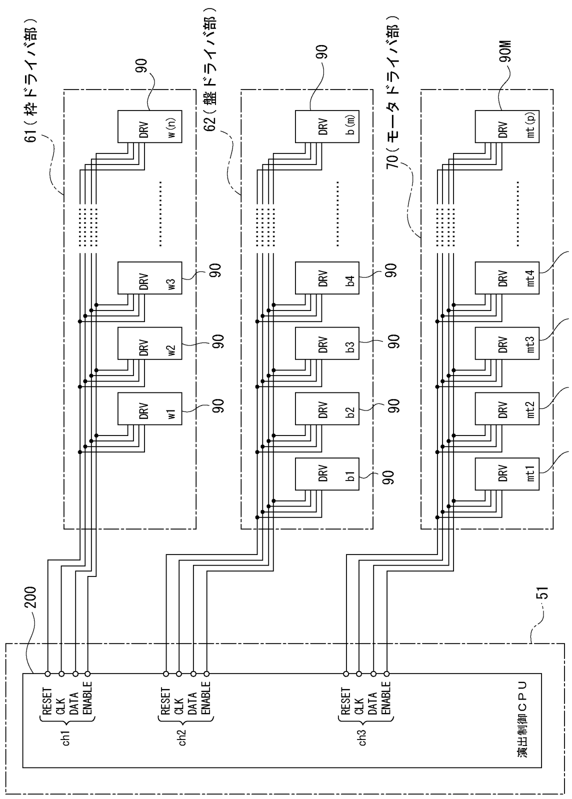
【図２】



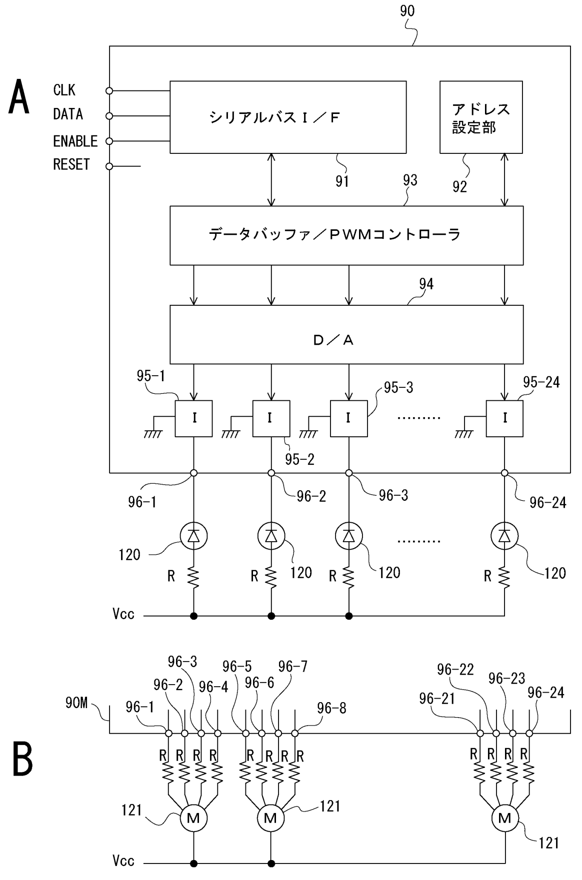
【図３】


1
50
主制御基板
CPU
100
ROM
101
RAM
102
71 上始動口センサ
72 下始動口センサ
73 ゲートセンサ
74 一般入賞口センサ
75 第１大入賞口センサ
76 第２大入賞口センサ
77 普通電動役物ソレノイド
78 第１大入賞口ソレノイド
79 第２大入賞口ソレノイド
80 第１特別図柄表示部
81 第２特別図柄表示部
82 普通図柄表示部
33 図柄表示部
53 払出制御基板
55 遊技球払出装置
54 発射制御基板
56 発射装置
57 枠用外部端子基板
66 液晶I/F基板
52 液晶制御基板
32 LCD
51 演出制御基板
200 CPU
201 ROM
202 RAM
ch1
61 枠ドライバ部
63 (20w) ランプ部
ch2
62 盤ドライバ部
64 (20b) ランプ部
ch3
70 モータドライバ部
可動体役物モータ
65
原点スイッチ
68
59 音源ＩＣ
67 アンプ部
25
69 音データＲＯＭ
60 操作部
58 電源基板
Vcc
外部電源 (AC24V)

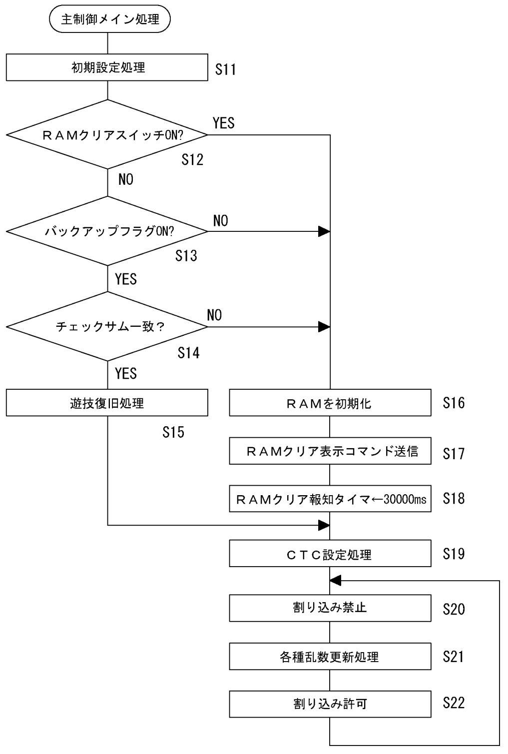
【図４】


61（枠ドライバ部）
62（盤ドライバ部）
70（モータドライバ部）
90
90M
DRV
w(n)
w3
w2
w1
b(m)
b4
b3
b2
b1
mt(p)
mt4
mt3
mt2
mt1
200
51
RESET
CLK
DATA
ENABLE
ch1
ch2
ch3
演出制御ＣＰＵ

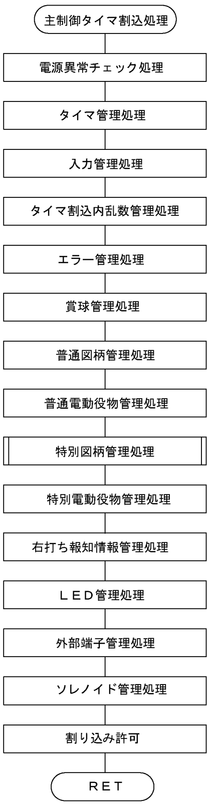
【図５】


A
90
CLK
DATA
ENABLE
RESET
シリアルバスＩ／Ｆ
アドレス設定部
91
93
92
データバッファ／ＰＷＭコントローラ
94
Ｄ／Ａ
95-1
95-2
95-3
95-24
I
96-1
96-2
96-3
96-24
120
R
Vcc
B
90M
96-1
96-2
96-3
96-4
96-5
96-6
96-7
96-8
96-21
96-22
96-23
96-24
R
M
121
Vcc

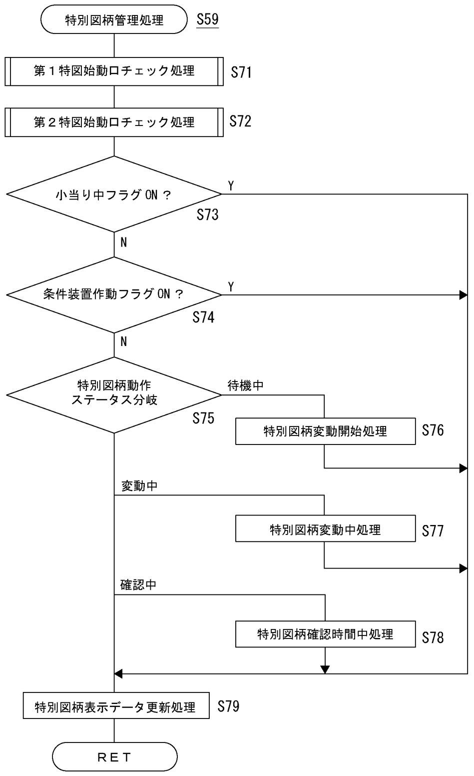
【図６】


主制御メイン処理
初期設定処理
S11
ＲＡＭクリアスイッチON?
S12
YES
NO
バックアップフラグON?
S13
NO
YES
チェックサム一致?
S14
NO
YES
遊技復旧処理
S15
ＲＡＭを初期化
S16
ＲＡＭクリア表示コマンド送信
S17
ＲＡＭクリア報知タイマ←30000ms
S18
ＣＴＣ設定処理
S19
割り込み禁止
S20
各種乱数更新処理
S21
割り込み許可
S22

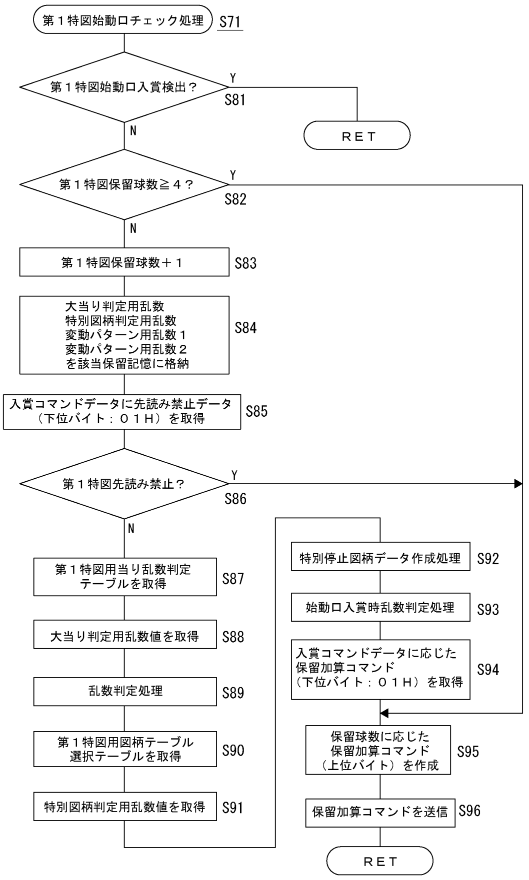
【図７】



【図８】



【図９】



【図１０】

B

図柄指定コマンド

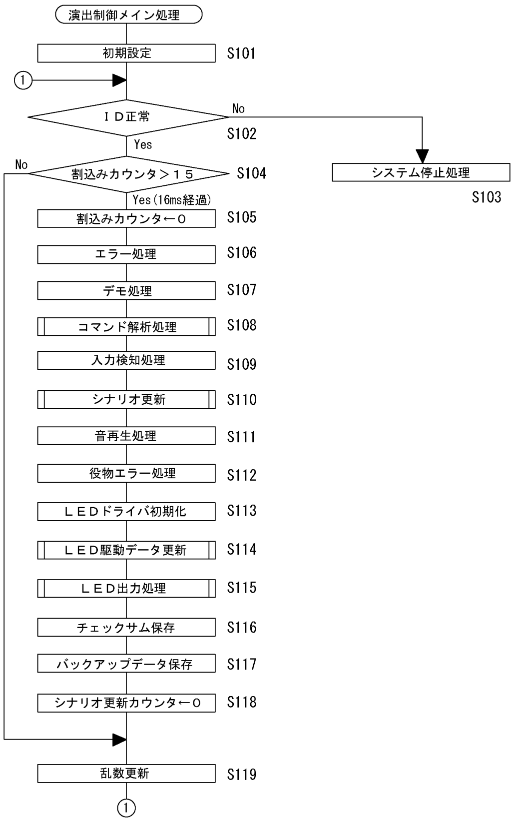
図柄	上位	下位	種類
第１特図	B5H	01H	外れ１
		02H	外れ２
		03H	外れ３
		04H	出玉多
		05H	出玉中
		06H	出玉少
第２特図	B6H	01H	外れ１
		02H	外れ２
		03H	外れ３
		04H	出玉多
		05H	出玉中
		06H	出玉少

A

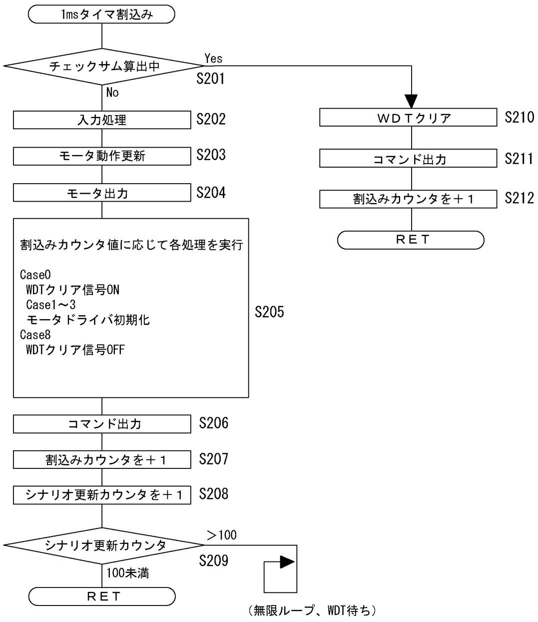
変動パターンコマンド、入賞時コマンド

種別	上位（入賞時）	上位（開始時）	下位	疑似	パターン
外れ	B0H	A0H	01H		通常変動５ｓ
			02H		通常変動８ｓ
			03H		通常変動12ｓ
			04H		ノーマルリーチ－１
			05H		ノーマルリーチ＋１
			06H	2	ノーマルリーチ＋１
			07H		スーパーリーチA
			08H	2	スーパーリーチA
			09H	3	スーパーリーチA
			0AH		スーパーリーチB
			0BH	2	スーパーリーチB
			0CH	3	スーパーリーチB
			0DH		スーパーリーチC
			0EH	2	スーパーリーチC
			0FH	3	スーパーリーチC
			10H	4	スーパーリーチC
			11H		ＳＴ中２ｓ
			12H		ＳＴ中８Ｓ
			13H		ＳＴ中リーチD
			14H		ＳＴ中リーチE
			15H		ＳＴ中リーチF
			16H		ＳＴ中リーチG
当り	B1H	A1H	01H		ノーマルリーチ
			02H	2	ノーマルリーチ
			03H		スーパーリーチA
			04H	2	スーパーリーチA
			05H	3	スーパーリーチA
			06H		スーパーリーチB
			07H	2	スーパーリーチB
			08H	3	スーパーリーチB
			09H		スーパーリーチC
			0AH	2	スーパーリーチC
			0BH	3	スーパーリーチC
			0CH	4	スーパーリーチC
			0DH	4	全回転
			0EH		ＳＴ中リーチD
			0FH		ＳＴ中リーチE
			10H		ＳＴ中リーチF
			11H		ＳＴ中リーチG

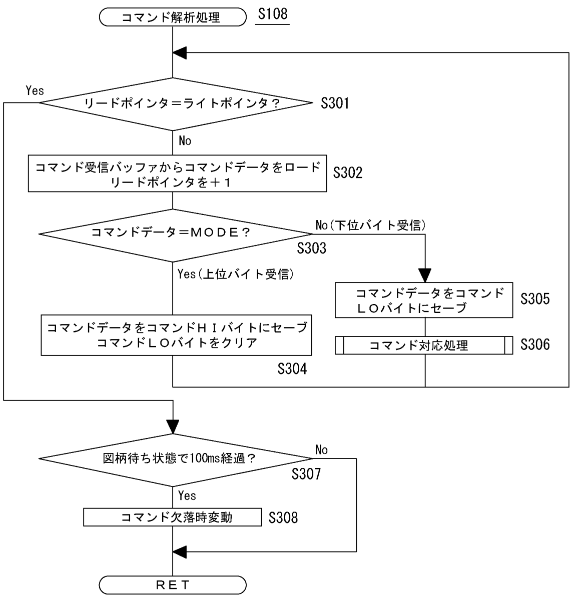
【図１１】



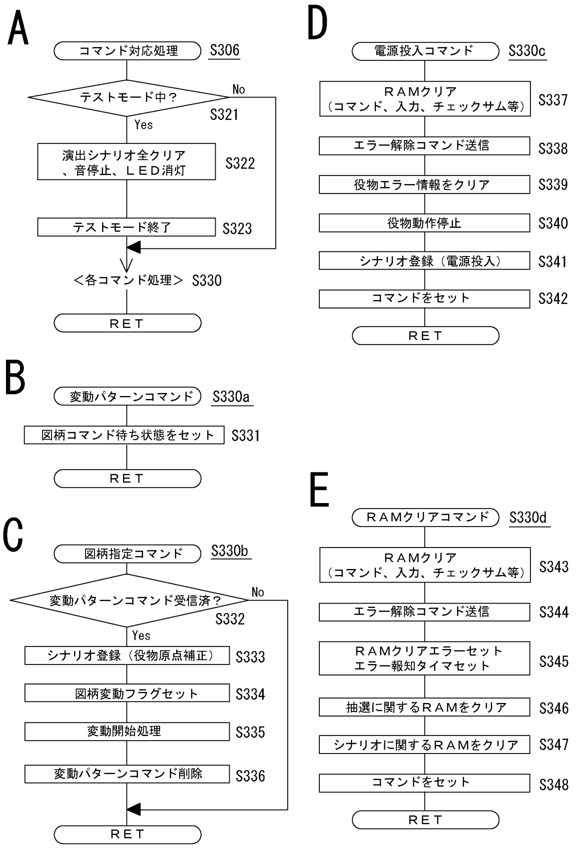
【図１２】



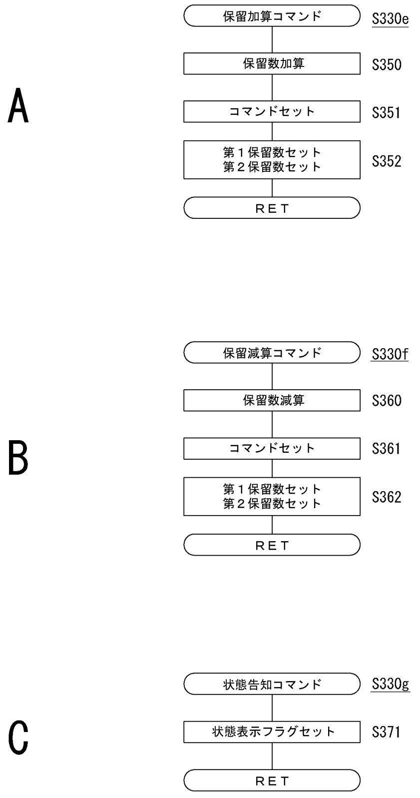
【図１３】



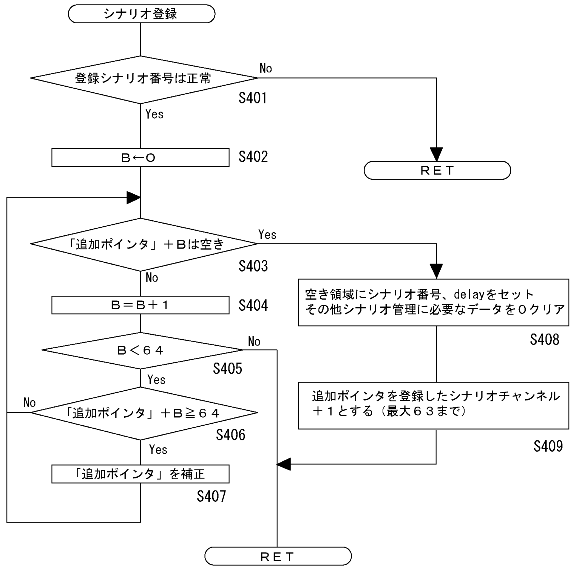
【図１４】



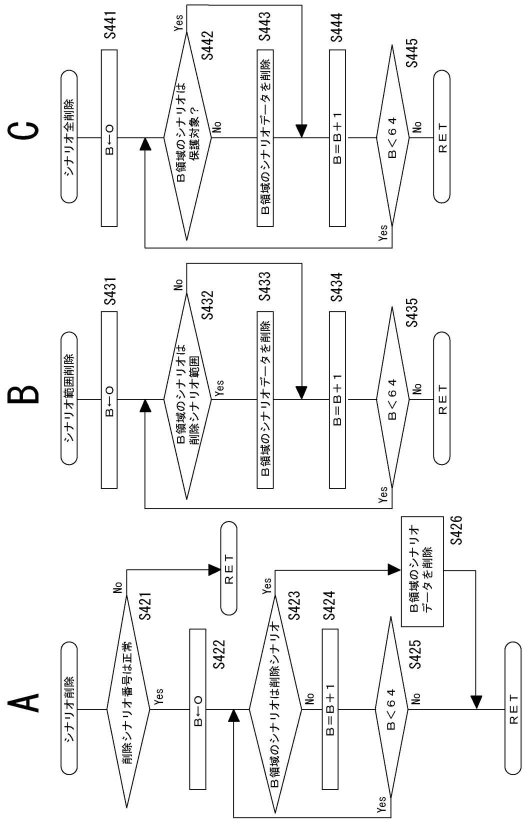
【図１５】



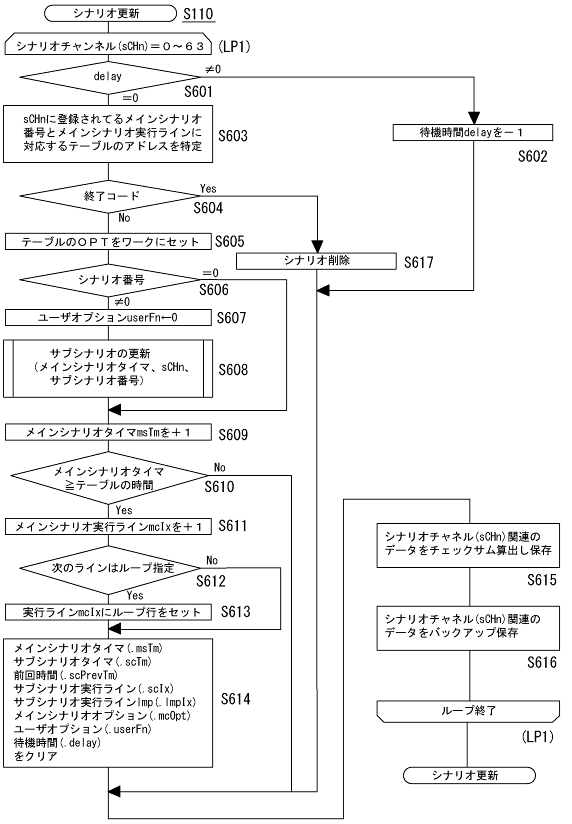
【図１６】



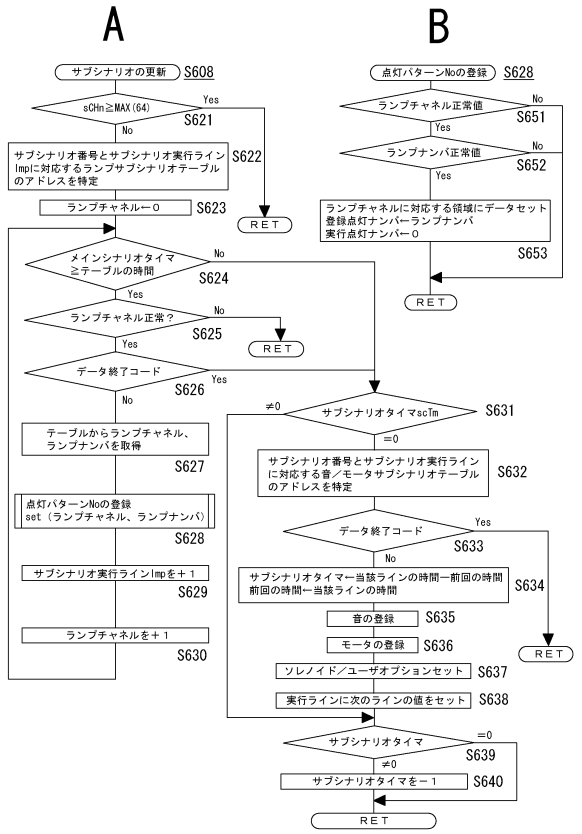
【図１７】



【図１８】



【図１９】



【図２０】

A

シナリオ登録情報（メイン、サブ）			シナリオチャネル（sCH）						
サイズ	名称	名称	0	1	2	・	・	・	63
DWORD	scTm;	サブシナリオタイマ							
DWORD	scPrevTm;	前回時間							
WORD	scIx;	サブシナリオ実行ライン							
WORD	lmpIx;	サブシナリオ実行ラインlmp							
DWORD	msTm;	メインシナリオタイマ							
WORD	mcIx;	メインシナリオ実行ライン							
WORD	mcNo;	メインシナリオ番号							
WORD	mcOpt;	メインシナリオオプション							
WORD	userFn;	ユーザオプション							
DWORD	delay;	待機時間							
DWORD	checkSum;	チェックサム							

B

ランプデータ登録情報			ランプチャネル（dwCH）						
サイズ	名称	名称	0	1	2	・	・	・	15
DWORD	lmpNew	登録点灯ナンバ							
DWORD	lmpNo	実行点灯ナンバ							
WORD	offset	実行ライン							
WORD	time	実行時間							
WORD	checksum	チェックサム							

C

モータデータ登録情報			モータチャネル（mCH）						
サイズ	名称	名称	0	1	2	・	・	・	7
BYTE	no;	実行動作ナンバ							
BYTE	noNew;	登録動作ナンバ							
BYTE	lcnt;	動作カウント(local)							
BYTE	tcnt;	励磁カウンタ							
WORD	step;	実行ステップ							
WORD	offset;	動作Line							
BYTE	attribute;	属性 親:0x5A　子:0x00							
BYTE	retNo;	親No							
BYTE	retAddr;	戻りアドレス							
WORD	roopAddr;	ループ開始ポイント							
WORD	roopCnt;	ループ回数							
BYTE	errCnt;	エラーカウンタ							
WORD	currentSw;	現在の入力情報							
WORD	softSw;	ソフト上のsw情報							
BYTE	softCnt;	ソフト上のカウント							

【図２１】

音データ登録情報			音チャネル（aCH）						
サイズ	名称	名称	0	1	2	・	・	・	15
BYTE	_frzVq:	ボリューム遷移量			0				
					1				
					1				
					0				
BYTE	_frzVl:	ボリューム			1		1		
					0		0		
					0		0		
					0		0		
BYTE	_rsv2:	遷移量変化	1		1		1		
	_rsv1:	ボリューム変化	1		1		1		
	_rsv0:	フレーズ変化	1		0		1		
	_frzSt:	ステレオ	1		1		0		
	_frzLp:	ループ	1						
	_frzHi:	フレーズ番号hi	0						
			0						
			0						
BYTE	_frzLo:	フレーズ番号low	0						
			0						
			1						
			1						
			1				1		
			1				1		
			1				0		
			1				1		

【図２２】

メインシナリオテーブル

＜シナリオ番号１＞

時間msTm	サブシナリオ番号scNo	OPT	
1	1		全体点滅
D_SELOP+0			

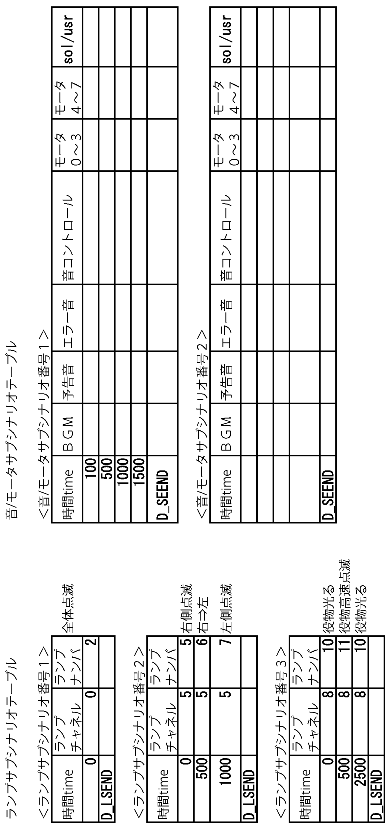
＜シナリオ番号２＞

時間msTm	サブシナリオ番号scNo	OPT	
1500	2		右点滅⇒左点滅
500	20		
2000	21		
D_SEEND			

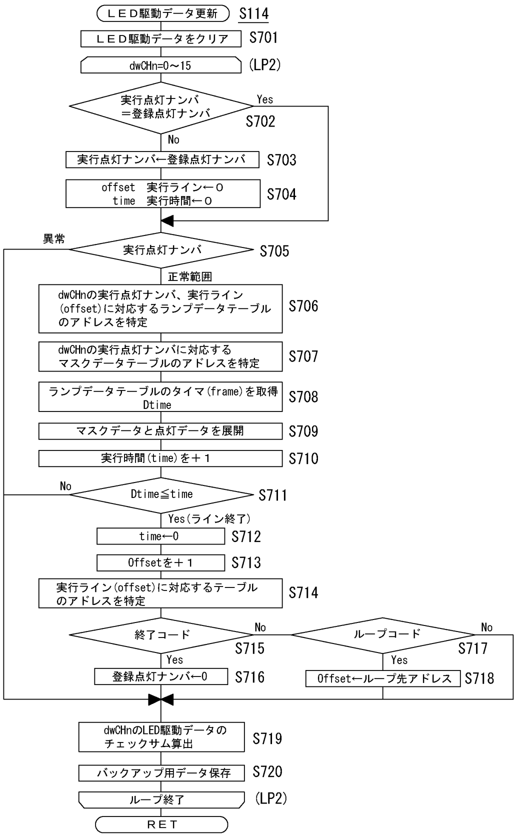
＜シナリオ番号３＞

時間msTm	サブシナリオ番号scNo	OPT	
3000	3		役物点滅、点滅、点灯
1000	25		
1000	26		
D_SEEND			

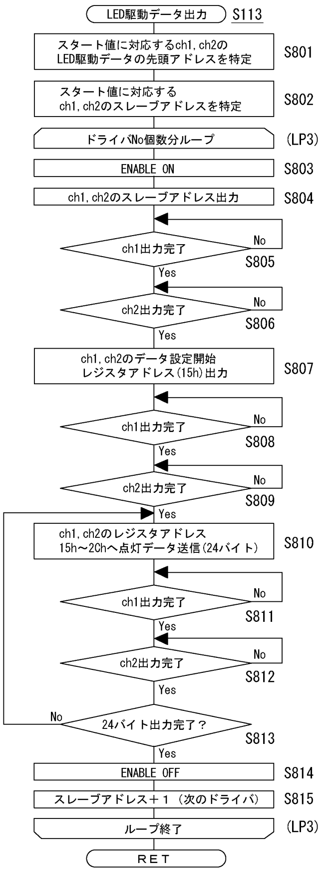
【図２３】



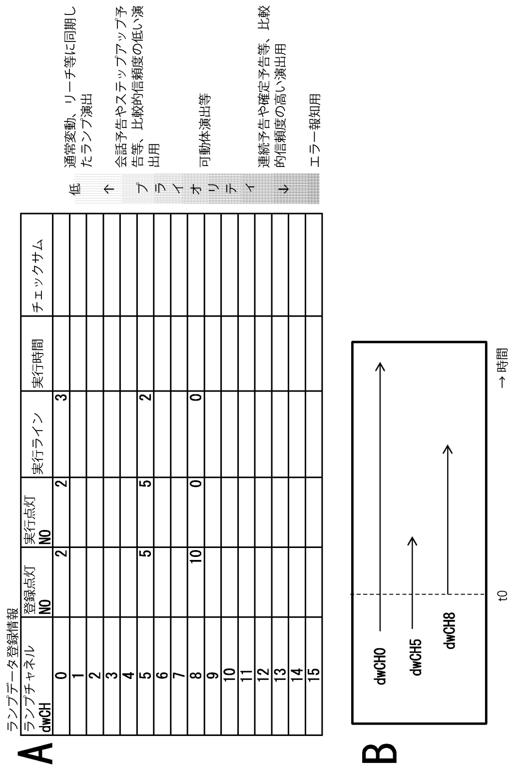
【図２４】



【図２５】



【図２６】

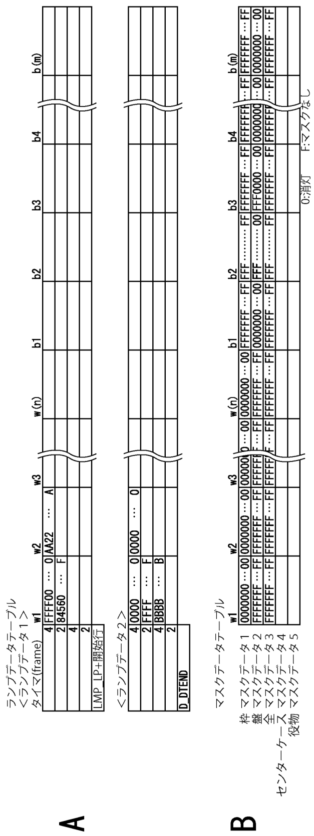


【図２７】

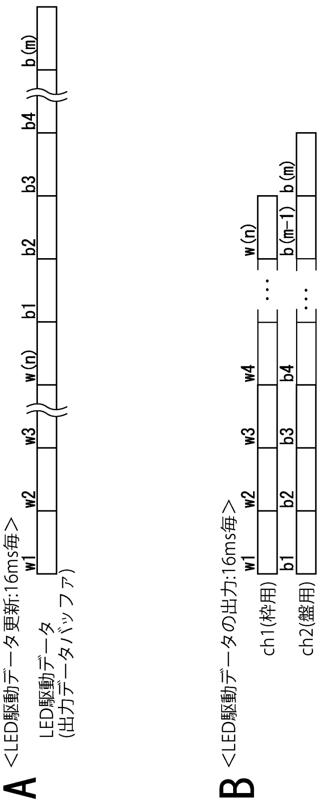
LampData_AddressTable

	ランプデータのアドレス	マスクデータのアドレス
0		
1		
2	全体点滅(通常BGM)	マスクなし
3		
4		
5	右側点滅	センターケースマスク
6	右⇒左	センターケースマスク
7	左側点滅	センターケースマスク
8		
9		
10	役物点灯	役物周辺をマスク
11	役物高速点滅	役物周辺をマスク
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		

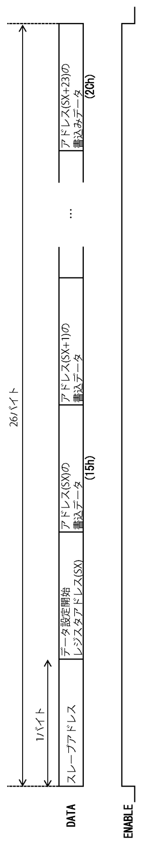
【図２８】



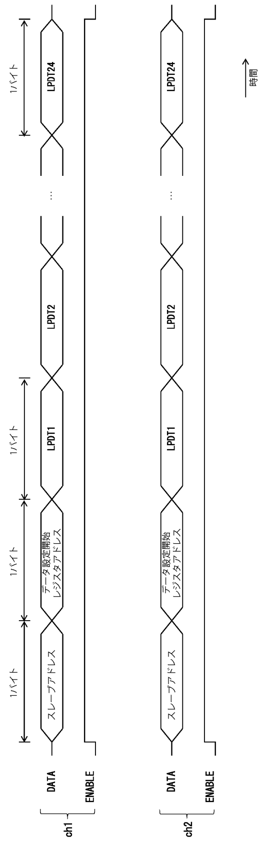
【図２９】



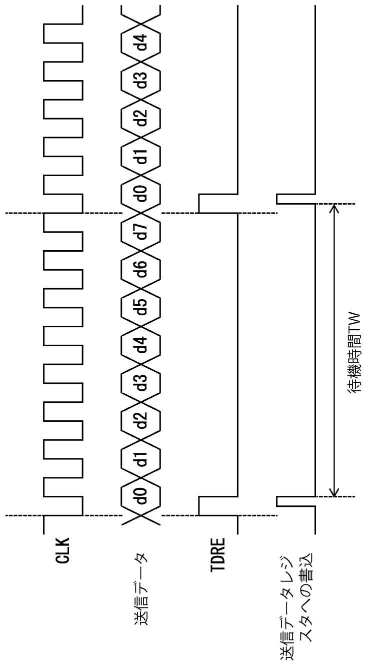
【図３０】



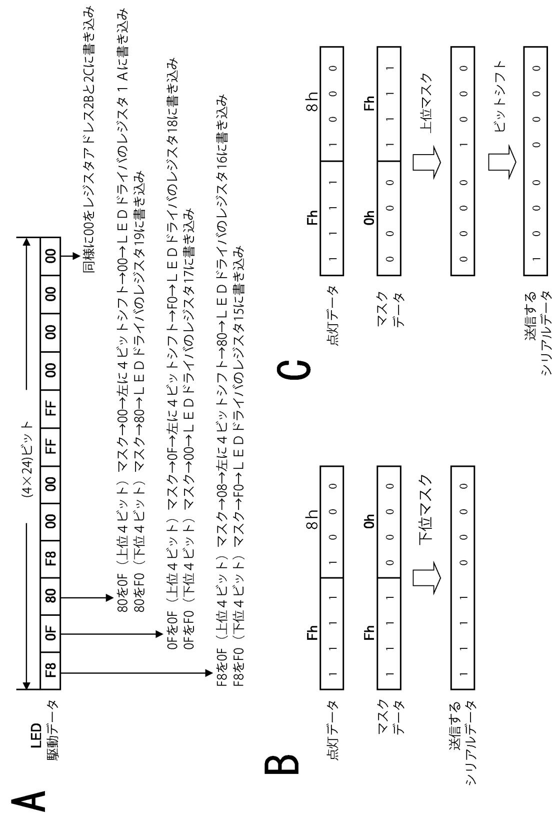
【図３１】



【図３２】



【図３３】



【図３４】

ランプデータテーブル
＜ランプデータ１＞

タイマ(frame)	w1	w2	w3	w(n)	b1	b2	b(m)
4	パターン１	パターン２		パターン１	パターン３		パターン５０
2	パターン１０	パターン１２		パターン２０	パターン５		パターンX
4	パターン５０	パターン６１		パターン８	パターン８		パターン４２
2	パターン２２	パターン２４		パターン９	パターン３２		パターンX
LMP_LP+開始行							

＜ランプデータ２＞

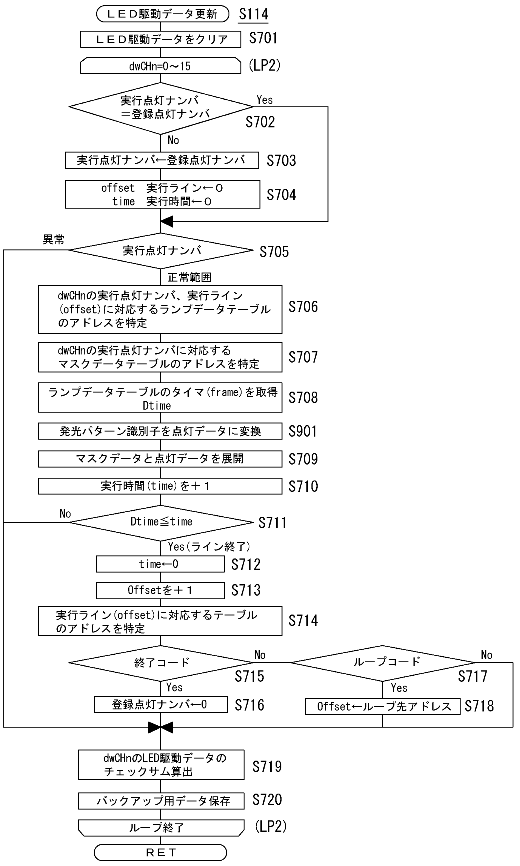
タイマ(frame)	w1	w2	w3	w(n)	b1	b2	b(m)
4	パターン１０１	パターン１０１		パターン１２０	パターン１２０		パターン１１９
2	パターン１５	パターン１２		パターン２４０	パターン２５０		パターンX
4	パターン２００	パターン２２２		パターン１２０	パターン１２０		パターン２５２
2	パターン１３０	パターン１３２		パターン６０	パターン７２		パターン８０
D_DTEND							

【図３５】

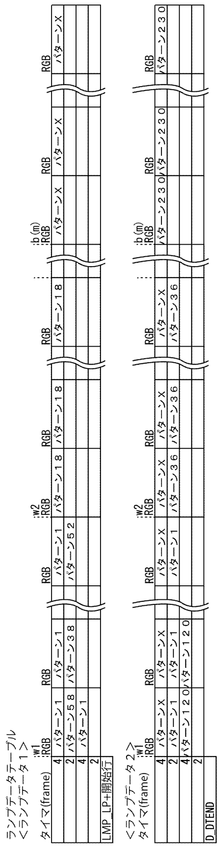
変換テーブル

点灯パターン識別子	点灯データ
パターン１	FFFF00・・・0
パターン２	AA22・・・A
・	・
・	・
・	・
パターンX	0000・・・F

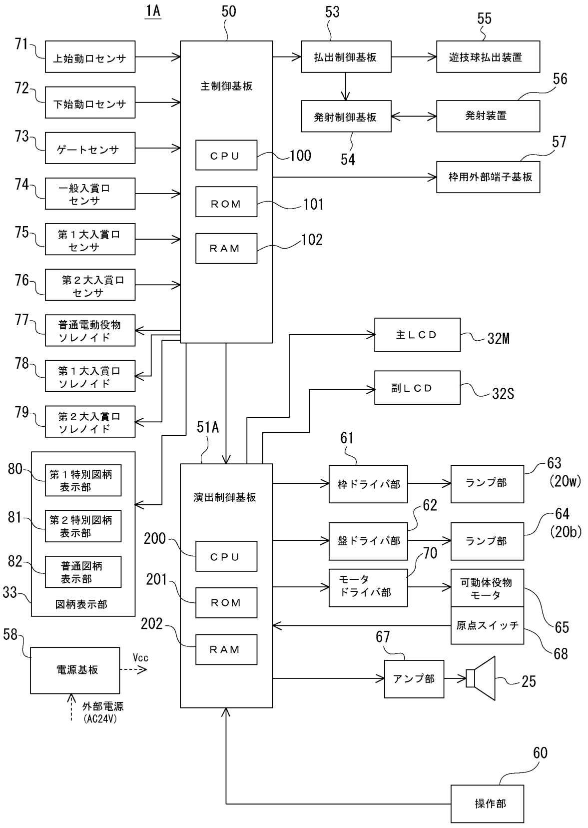
【図３６】



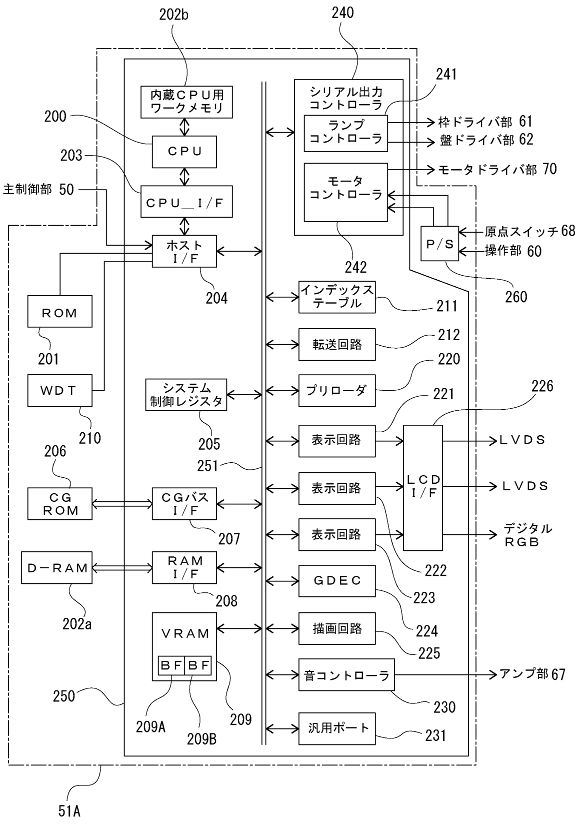
【図３７】



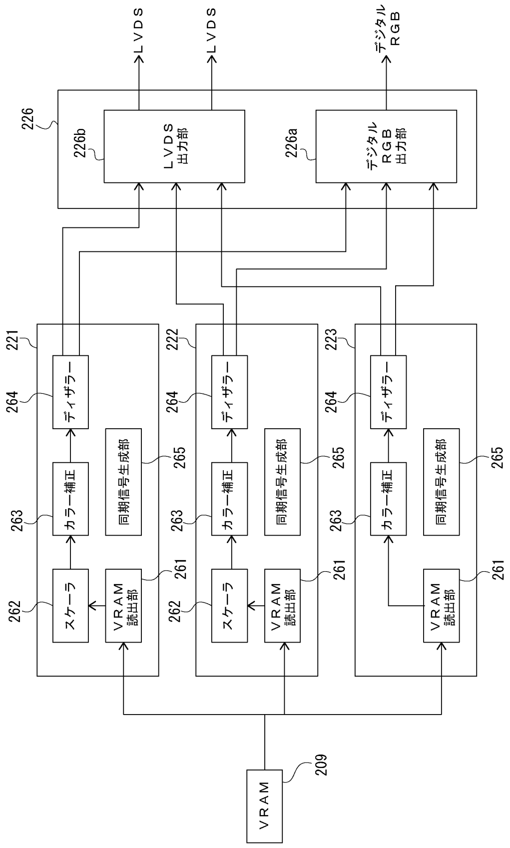
【図３８】



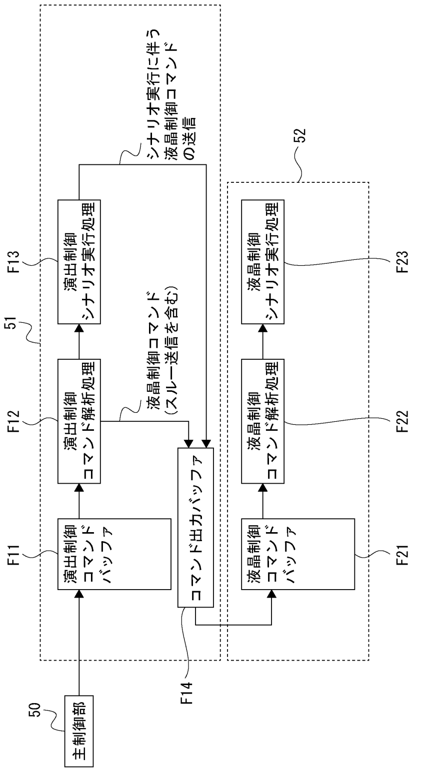
【図３９】



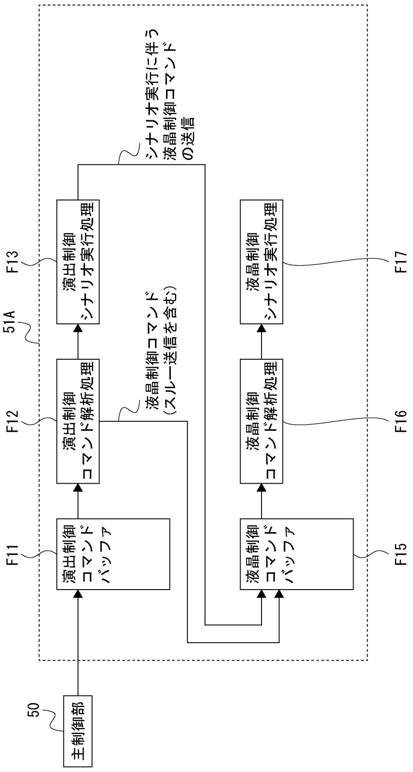
【図４０】



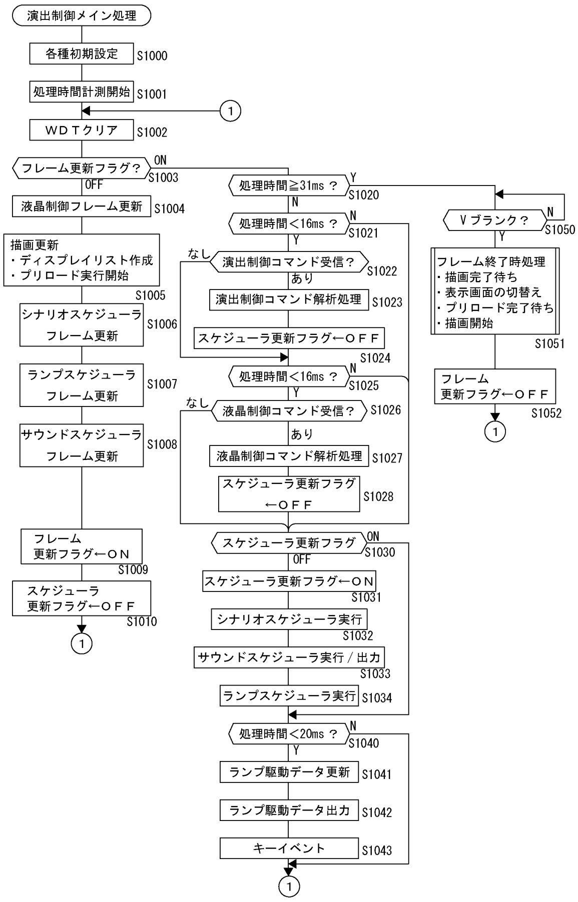
【図４１】



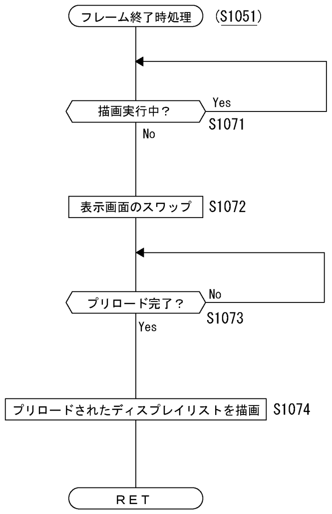
【図４２】



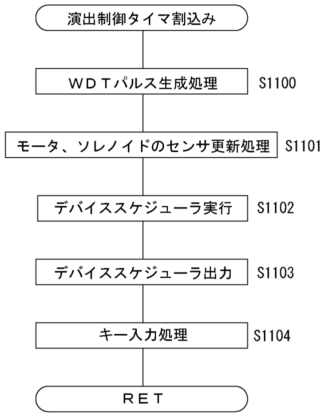
【図４３】



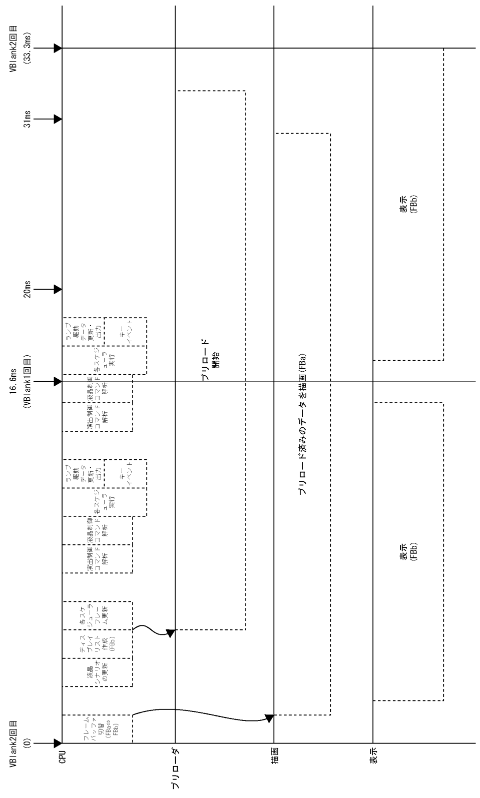
【図４４】



【図４５】



【図４６】



【図４７】

A

受信コマンド(上位)	コマンド名	対応処理
0x80	未使用	-
0x81	未使用	-
0x82	未使用	-
:	:	:
0xA0	特別図柄変動パターン	En_CommandAnalyze_Hendo
0xA1	特別図柄変動パターン	En_CommandAnalyze_Hendo
0xA2	特別図柄変動パターン	En_CommandAnalyze_Hendo
:	:	:
0xF0	大当り開始指定	En_CommandAnalyze_F0
0xF1	大入賞口開放回数指定	En_CommandAnalyze_F1
0xF2	大入賞口開放インターバル指定	En_CommandAnalyze_F2
→ 0xF3	大当り終了指定	En_CommandAnalyze_F3
:	:	:
0xFF	未使用	-

B

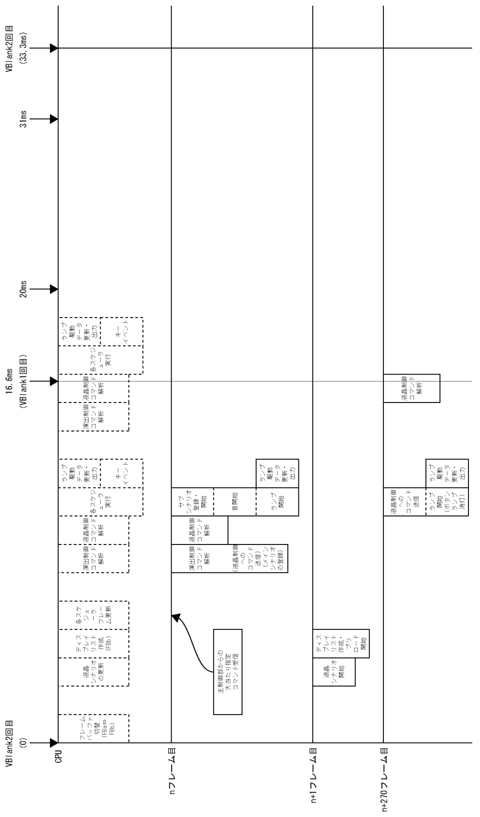
受信コマンド	液晶制御への送信コマンド	
F301H	F301H	スルー
	02xxH	選択キャラコマンド
	03xxH	連荘数コマンド
	05xxH	攻略フラグコマンド
	0AxxH	大当り図柄コマンド
	88xxH	大当り根幹コマンド
	9FxxH (9s後)	決定時のキャラコマンド
	01xxH (9s後)	背景指定コマンド

【図４８】

終了インターバルシナリオ

時間(f)	音	ランプ	モータ	ボタン	コマンド送信
0 (120)	音セット	ランプセット	モータセット		
120 (3)		ボタンランプセット			
123 (144)				有効開始	
267 (3)				有効終了	
270 (30)		ボタンランプ消灯			決定時のキャラコマンド(9FxxH)送信 背景指定コマンド(01xxH)送信
300	シナリオ終了				

【図４９】



フロントページの続き

(56)参考文献　特開２０１４－６８９０３（ＪＰ，Ａ）
特開２００９－６１１４７（ＪＰ，Ａ）
特開２０１４－２３１００２（ＪＰ，Ａ）
特開２０１７－１２２２５（ＪＰ，Ａ）
特開２００７－１０５３２５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ａ６３Ｆ　　７／０２