

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2020-5952
(P2020-5952A)

(43) 公開日 令和2年1月16日(2020.1.16)

(51) Int.Cl. F I テーマコード(参考)
A 6 3 F 5/04 (2006.01) A 6 3 F 5/04 5 1 2 Z 2 C 0 8 2
 A 6 3 F 5/04 5 1 2 D

審査請求 未請求 請求項の数 1 O L (全 66 頁)

(21) 出願番号 特願2018-130383 (P2018-130383)
 (22) 出願日 平成30年7月10日(2018.7.10)

(71) 出願人 598098526
 株式会社ユニバーサルエンターテインメント
 東京都江東区有明三丁目7番26号 有明
 フロントピアビルA棟
 (74) 代理人 110001520
 特許業務法人日誠国際特許事務所
 (72) 発明者 ▲高▼橋 和紀
 東京都江東区有明3丁目7番26号
 Fターム(参考) 2C082 AC23 AC31 AC51 AC61 AC81
 BB02 BB46 BB83 BB94 BB96
 CD32 CD41 CE18 DA52 DA54
 DA67 DA69 DA73

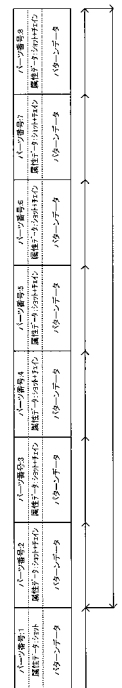
(54) 【発明の名称】遊技機

(57) 【要約】

【課題】発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供すること。

【解決手段】発光データが表すパーツ番号の順序にしたがってパーツデータに含まれるパターンデータでLEDを制御する。発光データにおける最後のパーツデータの属性データが連続再生(ショット+チェイン)であれば、属性データが連続再生を表す最初のパーツデータから発光データを再生する。遊技機がエラー状態から復帰した場合、遊技機がエラー状態となったときに再生していた発光データに対して属性データが連続再生を表していないパーツデータが除かれた演出データを再生する。

【選択図】図16



【特許請求の範囲】

【請求項 1】

複数の発光体により構成された発光部と、
 前記発光部の発光パターンを制御するための制御部と、
 前記発光部の前記発光パターンを制御するための複数の発光データ及び複数のパーツデータが記憶された発光データ記憶手段と、を備えた遊技機であって、
 前記パーツデータは、
 前記パーツデータを識別するための識別情報と、
 前記発光部の発光パターンを表すパターンデータと、
 前記パーツデータの属性を示す属性データと、を含み、
 前記発光データは、前記パーツデータの識別情報の順序を表し、
 前記制御部は、
 前記発光データ記憶手段に記憶された複数の発光データから、発光データを決定し、決定した発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、
 前記発光データにおける最後の識別情報に対応するパーツデータに含まれるパターンデータで前記発光部を制御した後、前記発光データにおける最後の識別情報に対応するパーツデータに含まれる前記属性データが連続再生を表すことを条件として、前記発光データが表す識別情報の先頭から前記属性データが連続再生を表すパーツデータを検索し、検出したパーツデータから前記発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、
 前記遊技機がエラー状態から復帰した場合、前記遊技機が前記エラー状態となったときに実行していた発光データが表すパーツデータの識別情報の順序から前記属性データが前記連続再生を表していないパーツデータの識別情報が除かれた発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御することを特徴とする遊技機。

10

20

【発明の詳細な説明】

【技術分野】

30

【0001】

本発明は、パチスロ等の遊技機に関する。

【背景技術】

【0002】

従来、複数の図柄がそれぞれの表面に配された複数のリールと、遊技メダルやコイン等（以下、「遊技媒体」という）が投入され、遊技者によりスタートレバーが操作されたことを検出し、複数のリールの回転の開始を要求するスタートスイッチと、複数のリールのそれぞれに対応して設けられたストップボタンが遊技者により押されたことを検出し、該当するリールの回転の停止を要求する信号を出力するストップスイッチと、複数のリールのそれぞれに対応して設けられ、それぞれの駆動力を各リールに伝達するステッピングモータと、スタートスイッチ及びストップスイッチにより出力された信号に基づいて、ステッピングモータの動作を制御し、各リールの回転及びその停止を行うリール制御装置とを備え、スタートレバーが操作されたことを検出すると、乱数値に基づいて抽籤を行い、この抽籤の結果（以下、「内部当籤役」という）とストップボタンが操作されたことを検出したタイミングとに基づいてリールの回転の停止を行う、いわゆるパチスロと称される遊技機が知られている。

40

【0003】

この種の遊技機として、LEDをドットマトリクス状に配置して表示ユニットを構成する遊技機が特許文献1に提案されている。

【先行技術文献】

50

【特許文献】

【0004】

【特許文献1】特開2005-323768号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

LEDをドットマトリクス状に配置したもののように、複数の発光体により構成された発光部の発光制御を行うためには、全ての発光パターンをROM等に記憶しておき、発光パターンを適宜更新する。このため、演出内容に応じた大量の発光パターンを用意しておく必要があり、その確認及び変更が開発者並びに確認者の負担となる。

10

【0006】

本発明は、このような課題を解決するためになされたもので、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供す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、

複数の発光体(LED)により構成された発光部(第1ランプ群111、第7ランプ群117、第8ランプ群118、ドットマトリクス部119)と、

前記発光部の発光パターンを制御するための制御部(サブCPU81)と

前記発光部の発光パターンを制御するための複数の発光データ及び複数のパーツデータが記憶された発光データ記憶手段(サブROM82)と、を備えた遊技機であって、

20

前記パーツデータは、

前記パーツデータを識別するための識別情報と、

前記発光部の発光パターンを表すパターンデータと、

前記パーツデータの属性を示す属性データと、を含み、

前記発光データは、前記パーツデータの識別情報の順序を表し、

前記制御部は、

前記発光データ記憶手段に記憶された複数の発光データから、発光データを決定し、決定した発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、

30

前記発光データにおける最後の識別情報に対応するパーツデータに含まれるパターンデータで前記発光部を制御した後、前記発光データにおける最後の識別情報に対応するパーツデータに含まれる前記属性データが連続再生を表すことを条件として、前記発光データが表す識別情報の先頭から前記属性データが連続再生を表すパーツデータを検索し、検出したパーツデータから前記発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、

前記遊技機がエラー状態から復帰した場合、前記遊技機が前記エラー状態となったときに実行していたエラー状態データが表すパーツデータの識別情報の順序から前記属性データが前記連続再生を表していないパーツデータの識別情報が除かれた発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御する

40

構成を有している。

【0008】

この構成により、本発明に係る遊技機は、パーツデータを組み合わせることによって、発光部に多様な演出を実行させるため、パーツデータ単位で発光部の発光パターンの確認及び変更を行わせることができる。このように、本発明に係る遊技機は、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【0009】

また、本発明に係る遊技機は、属性データが連続再生に設定されているパーツデータによって発光部に演出を繰り返し実行させるため、発光部による演出を長時間にわたって実

50

行させる場合であっても、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【0010】

また、本発明に係る遊技機は、エラー状態から復帰した場合、属性データが連続再生を表していないパーツデータの識別情報が除かれた新たな演出データが表す識別情報の順序にしたがってパーツデータに含まれるパターンデータで発光部を制御するため、エラー復帰時に演出データを最初から再生することにより遊技状態にそぐわない演出を実行してしまうことを防止することができる。

【発明の効果】

【0011】

本発明によれば、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供することができる。

【図面の簡単な説明】

【0012】

【図1】本発明の一実施形態の遊技機における機能フローを説明する説明図である。

【図2】本発明の一実施形態の遊技機における外観構成例を示す斜視図である。

【図3】本発明の一実施形態の遊技機におけるフロントパネルを外した状態の正面図である。

【図4】本発明の一実施形態の遊技機におけるLED配置ポート図である。

【図5】本発明の一実施形態の遊技機における内部構造を示すものであり、フロントドアを開いた状態の斜視図である。

【図6】本発明の一実施形態の遊技機が備える回路の全体構成を示すブロック図である。

【図7】本発明の一実施形態の遊技機における副制御回路の内部構成を示すブロック図である。

【図8】本発明の一実施形態の遊技機における第1モードの制御データのデータ構造を示す図である。

【図9】本発明の一実施形態の遊技機における第2モードの制御データのデータ構造を示す図である。

【図10】本発明の一実施形態の遊技機における第3モードの制御データのデータ構造を示す図である。

【図11】本発明の一実施形態の遊技機における図柄配置表の一例を示す図である。

【図12】本発明の一実施形態の遊技機における月別日テーブルを示す図である。

【図13】本発明の一実施形態の遊技機における日時格納領域を示す図である。

【図14】本発明の一実施形態の遊技機における再生状態管理格納領域を示す図である。

【図15】本発明の一実施形態の遊技機におけるランプデータが格納された再生状態管理格納領域の具体例を示す図である。

【図16】本発明の一実施形態の遊技機におけるランプデータの再生例を示す図である。

【図17】本発明の一実施形態の遊技機における中断復帰用ランプデータが格納された再生状態管理格納領域の具体例を示す図である。

【図18】本発明の一実施形態の遊技機における中断復帰用ランプデータの再生例を示す図である。

【図19】本発明の一実施形態の遊技機における主制御回路の電源投入処理を示すフローチャートである。

【図20】本発明の一実施形態の遊技機におけるメインCPUの割込処理を示すフローチャートである。

【図21】本発明の一実施形態の遊技機における副制御回路の電源投入処理を示すフローチャートである。

【図22】本発明の一実施形態におけるサブCPUにより行われる主基板通信タスクを示すフローチャートである。

【図23】本発明の一実施形態におけるサブCPUにより行われる演出登録タスクを示す

10

20

30

40

50

フローチャートである。

【図 2 4】本発明の一実施形態における演出内容決定処理を示すフローチャートである。

【図 2 5】本発明の一実施形態における無操作コマンド受信時処理の例を示すフローチャートである。

【図 2 6】本発明の一実施形態における日時更新処理を示すフローチャートである。

【図 2 7】本発明の一実施形態におけるサブ CPU により行われるサウンド制御タスクを示すフローチャートである。

【図 2 8】本発明の一実施形態におけるサウンド関数 RAM 転送処理を示すフローチャートである。

【図 2 9】本発明の一実施形態におけるサウンド制御タスク及びサウンド関数 RAM 転送処理の具体例をソースコードで表す概要図である。 10

【図 3 0】本発明の一実施形態におけるサブ CPU により行われるランプ制御タスクを示すフローチャートである。

【図 3 1】本発明の一実施形態におけるランプデータ読み込み処理を示すフローチャートである。

【図 3 2】本発明の一実施形態におけるランプ関数 RAM 転送処理を示すフローチャートである。

【図 3 3】本発明の一実施形態におけるランプ制御タスク、ランプデータ読み込み処理及びランプ関数 RAM 転送処理の具体例をソースコードで表す概要図である。

【図 3 4】本発明の一実施形態におけるサブ CPU に実行させるプログラムの製造方法を説明するための概念図である。 20

【発明を実施するための形態】

【0013】

以下、本発明の一実施形態を示す遊技機であるパチスロについて、図 1 ~ 図 3 4 を参照しながら説明する。

なお、本実施形態では、特定の図柄組合せが表示された場合にリプレイの当籤確率が通常時より高くなる遊技状態であるリプレイタイム（以下、「RT」という）が作動する機能を備えたパチスロについて説明する。

【0014】

<機能フロー>

まず、図 1 を参照して、パチスロの機能フローについて説明する。

本実施形態のパチスロでは、遊技を行うための遊技媒体としてメダルを用いる。なお、遊技媒体としては、メダル以外にも、コイン、遊技球、遊技用のポイントデータ又はトークン等を適用することもできる。

【0015】

遊技者によりメダルが投入され、スタートレバーが操作されると、予め定められた数値の範囲（例えば、0 ~ 6 5 5 3 5）の乱数から 1 つの値（以下、乱数値）が抽出される。

【0016】

内部抽籤手段は、抽出された乱数値に基づいて抽籤を行い、内部当籤役を決定する。この内部抽籤手段は、後述する主制御回路が担う。内部当籤役の決定により、後述の入賞判定ラインに沿って表示を行うことを許可する図柄の組合せが決定される。なお、図柄の組合せの種別としては、メダルの払い出し、再遊技の作動、ボーナスの作動等といった特典が遊技者に与えられる「入賞」に係るものと、それ以外のいわゆる「ハズレ」に係るものとが設けられている。 40

【0017】

また、スタートレバーが操作されると、複数のリールの回転が行われる。その後、遊技者により所定のリールに対応するストップボタンが押されると、リール停止制御手段は、内部当籤役とストップボタンが押されたタイミングとに基づいて、該当するリールの回転を停止する制御を行う。このリール停止制御手段は、後述する主制御回路が担う。

【0018】

パチスロでは、基本的に、ストップボタンが押されたときから規定時間（190 msec又は75 msec）内に、該当するリールの回転を停止する制御が行われる。本実施形態では、この規定時間内にリールの回転に伴って移動する図柄の数を「滑り駒数」と呼ぶ。規定期間が190 msecである場合には、滑り駒数の最大数を図柄4個分に定め、規定期間が75 msecである場合には、滑り駒数の最大数を図柄1個分に定める。

【0019】

リール停止制御手段は、入賞に係る図柄の組合せ表示を許可する内部当籤役が決定されているときは、通常、190 msec（図柄4コマ分）の規定時間内に、その図柄の組合せが入賞判定ラインに沿って極力表示されるようにリールの回転を停止させる。また、リール停止制御手段は、例えば、第2種特別役物であるチャレンジボーナス（CB）及びCBを連続して作動させるミドルボーナス（MB）の動作時には、1つ以上のリールに対して、規定時間75 msec（図柄1コマ分）内に、その図柄の組合せが入賞判定ラインに沿って極力表示されるようにリールの回転を停止させる。さらに、リール停止制御手段は、遊技状態に対応する各種規定時間を利用して、内部当籤役によってその表示が許可されていない図柄の組合せが入賞判定ラインに沿って表示されないようにリールの回転を停止させる。

【0020】

こうして、複数のリールの回転がすべて停止されると、入賞判定手段は、入賞判定ラインに沿って表示された図柄の組合せが、入賞に係るものであるか否かの判定を行う。この入賞判定手段は、後述する主制御回路が担う。入賞判定手段により入賞に係るものであるとの判定が行われると、メダルの払い出し等の特典が遊技者に与えられる。パチスロでは、以上のような一連の流れが1回の遊技として行われる。

【0021】

また、パチスロでは、前述した一連の流れの中で、表示装置により行う映像の表示、各種ランプにより行う光の出力、スピーカにより行う音の出力、或いはこれらの組合せを利用して様々な演出が行われる。

【0022】

スタートレバーが操作されると、上述した内部当籤役の決定に用いられた乱数値とは別に、演出用の乱数値（以下、演出用乱数値）が抽出される。演出用乱数値が抽出されると、演出内容決定手段は、内部当籤役に対応づけられた複数種類の演出内容の中から今回実行するものを抽籤により決定する。この演出内容決定手段は、後述する副制御回路が担う。

【0023】

演出内容が決定されると、演出実行手段は、リールの回転開始時、各リールの回転停止時、入賞の有無の判定時等の各契機に連動させて対応する演出を実行する。このように、パチスロでは、内部当籤役に対応づけられた演出内容を実行することによって、決定された内部当籤役（言い換えると、狙うべき図柄の組合せ）を知る機会又は予想する機会が遊技者に提供され、遊技者の興味の向上を図ることができる。

【0024】

<パチスロの構造>

次に、図2～図4を参照して、本実施形態におけるパチスロの構造について説明する。

【0025】

[外観構造]

図2は、パチスロ1の外部構造を示す斜視図である。図3は、本実施形態におけるパチスロ1のフロントパネル10を外した状態の正面図である。

【0026】

図2に示すように、パチスロ1は、外装体2を備えている。外装体2は、リールや回路基板等を収容するキャビネット2aと、キャビネット2aに対して開閉可能に取り付けられるフロントドア2bとを有している。

キャビネット2aの両側面には、把手7が設けられている（図2では一側面の把手7の

10

20

30

40

50

みを示す)。この把手7は、パチスロ1を運搬するときにかける凹部である。

【0027】

キャビネット2aの内部には、3つのリール3L, 3C, 3Rが横並びに設けられている。以下、各リール3L, 3C, 3Rを、それぞれ左リール3L、中リール3C、右リール3Rという。各リール3L, 3C, 3Rは、円筒状に形成されたリール本体と、リール本体の周面に装着された透光性のシート材と、リール本体の内側からシート材に光を照射するリール用光源とを有している。シート材の表面には、複数(例えば21個)の図柄が周方向に沿って所定の間隔をあけて描かれている。ここで、赤7図柄(不図示)の一部分は半透明部分とされている。赤また、リール本体には、シート材の背面に光を照射するリールバックライトが設けられている。このリールバックライトは、後述の副制御回路42

10

【0028】

フロントドア2bは、ドア本体9と、フロントパネル10と、発光表示装置11とを備えている。

ドア本体9は、ヒンジ(不図示)を用いてキャビネット2aに開閉可能に取り付けられている。ヒンジは、パチスロ1の前方からドア本体9を見た場合に、ドア本体9における左側の端部に設けられている。

【0029】

図2に示すように、発光表示装置11は、ドア本体9の上部に設けられている。この発光表示装置11は、マトリクス状に配置された複数の光源部によって形成されたドットマトリクス部119(図3参照)と、フロントパネル10のドットマトリクス部119に対向する部分から構成されている。

20

【0030】

ドットマトリクス部119は、任意の箇所の光源部を点灯(点滅)することで、フロントパネル10に施されたデザインの任意の箇所(本実施形態では花火の絵柄)を背面から照明する。これにより、フロントパネル10に施されたデザインの任意の箇所を発光させる演出が行われる。

【0031】

発光表示装置11の下方には、3つのリール3L, 3C, 3Rに描かれた図柄を表示する表示窓4L, 4C, 4Rが設けられている。以下、各表示窓4L, 4C, 4Rを、それぞれ左表示窓4L、中表示窓4C、右表示窓4Rという。

30

【0032】

表示窓4L, 4C, 4Rは、例えばアクリル板等の透明な部材で形成されている。この表示窓4L, 4C, 4Rは、正面(遊技者側)から見て、3つのリールの配置領域と重畳する位置に設けられ、かつ、3つのリールより手前(遊技者側)に位置するように設けられる。したがって、遊技者は、表示窓4L, 4C, 4Rを介して、表示窓4L, 4C, 4Rの背後に設けられた3つのリールを視認することができる。

【0033】

本実施形態では、表示窓4L, 4C, 4Rは、その背後に設けられた対応するリールの回転が停止したとき、各リールに描かれた複数種類の図柄のうち、連続して配置された3つの図柄を表示できる大きさに設定されている。すなわち、表示窓4L, 4C, 4Rの枠内には、リール毎に上段、中段及び下段の各領域が設けられ、各領域に1個の図柄が表示される。

40

【0034】

フロントパネル10は、ドア本体9の上部に取り付けられている。このフロントパネル10は、上表示部101と、リール照明部102と、リールサイド演出表示部103A, 103Bと、エッジ演出表示部104A, 104Bと、リール下表示部105とを有している。

【0035】

上表示部101は、発光表示装置11の上方に配置されており、リール照明部102は

50

、リール3L, 3C, 3Rと発光表示装置11との間に配置されている。リールサイド演出表示部103A, 103Bは、リール3L, 3C, 3Rの側方に配置されており、エッジ演出表示部104A, 104Bは、リール横演出表示部103の側方に配置されている。リール下表示部105は、リール3L, 3C, 3Rの下方に配置されている。

【0036】

上表示部101、リール照明部102、リールサイド演出表示部103A, 103B、エッジ演出表示部104A, 104B及びリール下表示部105は、ドア本体9に設けられた後述する各種ランプ群111~117を覆っている。そして、これら上表示部101, 102, 103A, 103B, 104A, 104B, 105は、各種ランプ群111~117からの光が照射されて発光する。

10

【0037】

例えば、リールサイド演出表示部103Aには、上下方向に並ぶ3つのBET発光部が設けられている。3つのBET発光部の点灯する数は、1回の遊技に使用するメダルを示す。

【0038】

本実施形態では、1回の遊技に使用するメダルの数を1~3に設定している。例えば、1回の遊技に使用するメダルの数を「1」にした場合は、1つのBET発光部(例えば一番下に位置するBET発光部)が点灯し、その他BET発光部(真ん中と一番上に位置するBET発光部)が消灯する。

【0039】

図2に示すように、ドア本体9の中央には、台座部12が形成されている。この台座部12には、遊技者の操作対象となる各種装置(メダル投入口13、MAXベットボタン14、1ベットボタン15、スタートレバー16、ストップボタン17L, 17C, 17R)が設けられている。

20

【0040】

メダル投入口13は、遊技者によって外部からパチスロ1に投下されるメダルを受け入れるために設けられる。メダル投入口13から受け入れられたメダルは、予め設定された枚数(例えば3枚)を上限として1回の遊技に使用され、予め設定された枚数を越えた分は、パチスロ1の内部に預けることができる(いわゆるクレジット機能)。

【0041】

MAXベットボタン14及び1ベットボタン15は、パチスロ1の内部に預けられているメダルから1回の遊技に使用する枚数を決定するために設けられる。なお、図2には示さないが、台座部12には、精算ボタンが設けられる。この精算ボタンは、パチスロ1の内部に預けられているメダルを外部に引き出す(排出する)ために設けられる。

30

【0042】

スタートレバー16は、全てのリール(3L, 3C, 3R)の回転を開始するために設けられる。ストップボタン17L, 17C, 17Rは、それぞれ、左リール3L、中リール3C、右リール3Rに対応づけて設けられ、各ストップボタンは対応するリールの回転を停止するために設けられる。以下、ストップボタン17L, 17C, 17Rを、それぞれ左ストップボタン17L、中ストップボタン17C、右ストップボタン17Rという。

40

【0043】

また、台座部12には、7セグメントLED(Light Emitting Diode)からなる7セグ表示器6が設けられている。この7セグ表示器6は、特典として遊技者に対して払い出すメダルの枚数(以下、払出枚数)、パチスロ1の内部に預けられているメダルの枚数(以下、クレジット枚数)、遊技を行うためのメダルの投入枚数(以下、BET枚数)等の情報をデジタル表示する。

【0044】

ドア本体9の下部には、メダル払出口18、メダル受皿19、スピーカ20L, 20R等が設けられている。メダル払出口18は、後述のメダル払出装装置33の駆動により排出されるメダルを外部に導く。メダル受皿19は、メダル払出口18から排出されたメダル

50

を貯める。また、スピーカ 20L, 20R は、演出内容に対応する効果音や楽曲等の音を出力する。

【0045】

また、ドア本体 9 には、腰部パネル表示部 106 が設けられている。この腰部パネル表示部 106 は、スピーカ 20L, 20R の上方に配置されている。腰部パネル表示部 106 は、ドア本体 9 に設けられた後述する光源部 (No. 53 のポート) を覆っている。そして、腰部パネル表示部 106 は、光源部 (No. 53 のポート) からの光が照射されて発光する。

【0046】

[各種ランプ群]

図 4 は、パチスロ 1 における LED 配置ポート図である。

【0047】

図 4 に示すように、ドア本体 9 には、337 ポートの光源部が設けられている。各光源部には、発光体として少なくとも 1 つの LED が配置されている。なお、各光源部には、有機エレクトロルミネッセンス等の他の発光体を配置してもよい。

【0048】

No. 0 ~ No. 30 のポートに係る光源部は、第 1 ランプ群 111 を形成している。この第 1 ランプ群 111 は、上表示部 101 (図 3 参照) に光を照射する。No. 49 ~ No. 52 のポートに係る光源部は、第 2 ランプ群 112 を形成している。この第 2 ランプ群 112 は、リール照明部 102 (図 3 参照) に光を照射する。

【0049】

No. 66 ~ No. 70 のポートに係る光源部は、第 3 ランプ群 113 を形成し、No. 71 ~ No. 75 のポートに係る光源部は、第 4 ランプ群 114 を形成している。第 3 ランプ群 113 は、リールサイド演出表示部 103A (図 3 参照) に光を照射し、第 4 ランプ群 114 は、リールサイド演出表示部 103B (図 3 参照) に光を照射する。第 3 ランプ群 113 の No. 68 ~ No. 70 のポートに係る光源部は、上述した 3 つの BET 発光部に対向し、点灯することでそれぞれ対向する BET 発光部を発光させる。

【0050】

No. 54 ~ No. 59 のポートに係る光源部は、第 5 ランプ群 115 を形成し、No. 60 ~ No. 65 のポートに係る光源部は、第 6 ランプ群 116 を形成している。第 5 ランプ群 115 は、エッジ演出表示部 104A (図 3 参照) に光を照射し、第 6 ランプ群 116 は、エッジ演出表示部 104B (図 3 参照) に光を照射する。

【0051】

No. 108 ~ No. 156 のポートに係る光源部は、第 7 ランプ群 117 を形成している。第 7 ランプ群 117 は、リール下表示部 105 (図 3 参照) に光を照射する。第 7 ランプ群 117 の No. 108 ~ No. 121 のポートに係る光源部は、例えば、パチスロ 1 の内部に預けられているメダルの枚数である貯留枚数を表示する。

【0052】

第 7 ランプ群 117 の No. 122 ~ No. 142 のポートに係る光源部は、例えば、ボーナス中のメダルの獲得枚数を表示する。第 7 ランプ群 117 の No. 143 ~ No. 156 のポートに係る光源部は、例えば、メダルの払出枚数を表示する。

【0053】

No. 31 ~ No. 48 及び No. 76 ~ No. 93 のポートに係る光源部は、第 8 ランプ群 118 を形成している。第 8 ランプ群 118 の No. 31 ~ No. 36 のポートに係る光源部は、左リール 3L のリール用光源 (リールバックライト) として機能する。また、No. 37 ~ No. 42 のポートに係る光源部は、中リール 3C のリール用光源 (リールバックライト) として機能し、No. 43 ~ No. 48 のポートに係る光源部は、右リール 3R のリール用光源 (リールバックライト) として機能する。

【0054】

No. 53 のポートに係る光源部は、腰部パネル表示部 106 (図 3 参照) に光を照射

10

20

30

40

50

する。No.157のポートに係る光源部は、MAXBETボタン14に光を照射する。左ストップボタン17Lに光を照射する。No.158のポートに係る光源部は、左ストップボタン17Lに光を照射する。No.159のポートに係る光源部は、中ストップボタン17Cに光を照射する。No.160のポートに係る光源部は、右ストップボタン17Rに光を照射する。

【0055】

No.161~No.337のポートに係る光源部は、ドットマトリクス部119を形成している。このドットマトリクス部119は、発光表示装置11の光源として機能する。ドットマトリクス部119の各ポートには、1個のLEDが設けられている。

【0056】

[内部構造]

次に、パチスロ1の内部構造を、図5を参照しながら説明する。図5は、パチスロ1の内部構造を示す斜視図である。

【0057】

キャビネット2aは、正面側の一面が開口された略直方体状に形成されている。このキャビネット2a内の上部には、後述の主制御回路41(図6参照)を構成する主基板31が設けられている。主制御回路41は、内部当籤役の決定、各リールの回転及び停止、入賞の有無の判定等の、パチスロ1における遊技の主な動作及び該動作間の流れを制御する回路である。なお、主制御回路41の具体的な構成は後述する。

【0058】

キャビネット2a内の中央部には、3つのリール(左リール3L、中リール3C及び右リール3R)が設けられている。なお、図5には示さないが、各リールは、所定の減速比を有する歯車を介して対応する後述のステッピングモータ(図6中のステッピングモータ61L,61C,61Rのいずれか)に接続される。

【0059】

キャビネット2a内の下部には、多量のメダルを収容可能であり、かつ、それらを1枚ずつ排出可能な構造を有するメダル払出装装置33(以下、ホッパー33という)が設けられている。また、キャビネット2a内における、ホッパー33の一方の側部(図5に示す例では左側)には、パチスロ1が有する各装置に対して必要な電力を供給する電源装置34が設けられている。

【0060】

フロントドア2bの裏面側(表示画面側とは反対側の部分)における上部には、後述の副制御回路42(図6及び図7参照)を構成する副基板32が設けられている。副制御回路42は、映像の表示等による演出の実行を制御する回路である。なお、副制御回路42の具体的な構成は後述する。

【0061】

さらに、フロントドア2bの裏面側における略中央部には、セレクトア35が設けられている。セレクトア35は、メダル投入口13(図2参照)を介して外部から投入されたメダルの材質や形状等が適正である否かを選別する装置であり、適正であると判定したメダルをホッパー33に案内する。また、図5には示さないが、セレクトア35内においてメダルが通過する経路上には、適正なメダルが通過したことを検出するメダルセンサ35S(図6参照)が設けられている。

【0062】

<パチスロ1が備える回路の構成>

次に、パチスロ1が備える回路の構成について、図6及び図7を参照して説明する。

図6は、パチスロ1が備える回路全体のブロック構成図である。図7は、副制御回路の内部構成を示すブロック構成図である。

【0063】

パチスロ1は、主制御回路41、副制御回路42、及び、これらの回路と電氣的に接続される周辺装置(アクチュエータ)を備える。

10

20

30

40

50

【 0 0 6 4 】

[主制御回路]

主制御回路 4 1 は、主に、回路基板（主基板 3 1）上に設置されたマイクロコンピュータ 5 0 により構成される。それ以外の構成要素として、主制御回路 4 1 は、クロックパルス発生回路 5 4、分周器 5 5、乱数発生器 5 6、サンプリング回路 5 7、表示部駆動回路 6 4、ホッパー駆動回路 6 5、及び、払出完了信号回路 6 6 を含む。

【 0 0 6 5 】

マイクロコンピュータ 5 0 は、メイン CPU 5 1、メイン ROM（Read Only Memory）5 2 及びメイン RAM（Random Access Memory）5 3 により構成される。

【 0 0 6 6 】

メイン ROM 5 2 には、メイン CPU 5 1 により実行される各種処理の制御プログラム、内部抽籤テーブル等のデータテーブル、副制御回路 4 2 に対して各種制御指令（コマンド）を送信するためのデータ等が記憶されている。メイン RAM 5 3 には、制御プログラムの実行により決定された内部当籤役等の各種データを格納する格納領域が設けられている。

10

【 0 0 6 7 】

メイン CPU 5 1 には、クロックパルス発生回路 5 4、分周器 5 5、乱数発生器 5 6 及びサンプリング回路 5 7 が接続されている。クロックパルス発生回路 5 4 及び分周器 5 5 は、クロックパルスを発生する。なお、メイン CPU 5 1 は、発生されたクロックパルスに基づいて、制御プログラムを実行する。また、乱数発生器 5 6 は、予め定められた範囲の乱数（例えば、0 ~ 6 5 5 3 5）を発生する。そして、サンプリング回路 5 7 は、発生された乱数の中から 1 つの値を抽出する。

20

【 0 0 6 8 】

マイクロコンピュータ 5 0 の入力ポートには、各種スイッチ及びセンサ等が接続される。メイン CPU 5 1 は、各種スイッチ等からの入力信号を受けて、ステッピングモータ 6 1 L、6 1 C、6 1 R 等の周辺装置の動作を制御する。

【 0 0 6 9 】

ストップスイッチ 1 7 S は、本発明に係る停止操作検出手段の一具体例を示すものであり、左ストップボタン 1 7 L、中ストップボタン 1 7 C、右ストップボタン 1 7 R のそれぞれが遊技者により押されたこと（停止操作）を検出する。スタートスイッチ 1 6 S は、本発明に係る開始操作検出手段の一具体例を示すものであり、スタートレバー 1 6 が遊技者により操作されたこと（開始操作）を検出する。精算スイッチ 1 4 S は、精算ボタンが遊技者により押されたことを検出する。

30

【 0 0 7 0 】

メダルセンサ 3 5 S は、本発明に係る投入操作検出手段の一具体例を示すものであり、メダル投入口 1 3 に投入されたメダルがセレクタ 3 5 内を通過したことを検出する。また、ベットスイッチ 1 2 S は、ベットボタン（MAXベットボタン 1 4 又は 1ベットボタン 1 5）が遊技者により押されたことを検出する。

【 0 0 7 1 】

また、マイクロコンピュータ 5 0 により動作が制御される周辺装置としては、3つのステッピングモータ 6 1 L、6 1 C、6 1 R、7セグ表示器 6 及びホッパー 3 3 がある。また、マイクロコンピュータ 5 0 の出力ポートには、各周辺装置の動作を制御するための駆動回路が接続される。

40

【 0 0 7 2 】

モータ駆動回路 6 2 は、左リール 3 L、中リール 3 C、右リール 3 R に対応してそれぞれ設けられた3つのステッピングモータ 6 1 L、6 1 C、6 1 R の駆動を制御する。リール位置検出回路 6 3 は、センサ発光部とセンサ受光部とを有する光センサにより、リールが一回転したことを示すリールインデックスをリール毎に検出する。

【 0 0 7 3 】

3つのステッピングモータ 6 1 L、6 1 C、6 1 R のそれぞれは、その運動量がパルス

50

の出力数に比例し、回転軸を指定された角度で停止させることが可能な構成を有する。また、各ステッピングモータの駆動力は、所定の減速比を有する歯車を介して、対応するリールに伝達される。そして、各ステッピングモータに対して1回のパルスが出力されるごとに、対応するリールは一定の角度で回転する。3つのリール3L、3C、3R及び3つのステッピングモータ61L、61C、61Rは、本発明に係る変動表示手段の一具体例を示すものである。

【0074】

メインCPU51は、各リールのリールインデックスを検出してから対応するステッピングモータに対してパルスが出力された回数をカウントすることによって、各リールの回転角度（具体的には、リールが図柄何個分だけ回転したか）を管理する。

10

【0075】

ここで、各リールの回転角度の管理を具体的に説明する。各ステッピングモータに対して出力されたパルスの数は、メインRAM53に設けられたパルスカウンタ（不図示）によって計数される。そして、図柄1個分の回転に必要な所定回数（例えば16回）のパルスの出力がパルスカウンタで計数されるごとに、メインRAM53に設けられた図柄カウンタ（不図示）の値に、「1」が加算される。なお、図柄カウンタは、リール毎に設けられる。そして、図柄カウンタの値は、リール位置検出回路63によってリールインデックスが検出されるとクリアされる。

【0076】

すなわち、本実施形態では、図柄カウンタの値を管理することにより、リールインデックスが検出されてから図柄何個分の回転動作が行われたのかを管理する。それゆえ、各リールの各図柄の位置は、リールインデックスが検出される位置を基準として検出される。

20

【0077】

なお、表示部駆動回路64は、7セグ表示器6の動作を制御する。ホッパー駆動回路65は、ホッパー33の動作を制御する。払出完了信号回路66は、ホッパー33に設けられたメダル検出部33Sが行うメダルの検出を管理し、ホッパー33から外部に排出されたメダルが所定の払出枚数に達したか否かをチェックする。また、主制御回路41には、外部端子板18Sが接続されている。主制御回路41は、外部端子板18Sを介してホールコンピュータ又は呼出装置100に接続されている。

【0078】

30

[副制御回路]

図7に示すように、副制御回路42は、主制御回路41と電氣的に接続され、主制御回路41から送信されるコマンドに基づいて演出内容の決定や実行等の処理を行う。副制御回路42は、サブCPU81、サブROM82、サブRAM83（演算記憶手段）、発振回路84、バックアップRAM85、RTC（Real-Time Clock）86、バッテリー87及びサウンドIC88を含んで構成される。

【0079】

サブCPU81は、主制御回路41から送信されたコマンドに応じて、サブROM82に記憶されている制御プログラムに従い、音、光の出力制御を行う。サブCPU81は、制御部及び演算処理手段を構成する。

40

【0080】

サブROM82は、本発明の遊技機に係る記憶手段の一具体例を示すものであり、基本的には、プログラム記憶領域及びデータ記憶領域を有する。このように、サブROM82は、サブCPU81を動作させるためのプログラム及びデータが記憶された第1記憶手段を構成する。

【0081】

プログラム記憶領域には、サブCPU81が実行する各種制御プログラムが記憶される。なお、プログラム記憶領域に格納される制御プログラムには、例えば、主制御回路41との通信を制御するための主基板通信タスク、演出用乱数値を抽出して演出内容（演出データ）の決定及び登録を行うための演出登録タスク、決定した演出内容に基づいて発光表

50

示装置 11 のドットマトリクス部 119 及び各種ランプ群による光の出力を制御するためのランプ制御タスク、スピーカ 20L, 20R による音の出力を制御するためのサウンド制御タスク等のプログラムが含まれる。

【0082】

データ記憶領域には、例えば、各種データテーブルを記憶する記憶領域、各種演出内容を構成する演出データを記憶する記憶領域、BGM や効果音に関するサウンドデータを記憶する記憶領域、光の点消灯のパターンに関するランプデータを記憶する記憶領域等の各種記憶領域が含まれる。なお、ランプデータについては、図 14 ~ 図 18 を参照して後述する。

【0083】

サブRAM 83 は、サブROM 82 よりもアクセス速度が高く、かつ、書き換え可能な第 2 記憶手段を構成し、例えば、DRAM (Dynamic Random Access Memory)、又は、SRAM (Static Random Access Memory) によって構成されている。

【0084】

サブRAM 83 は、決定された演出内容や演出データを登録する格納領域や、主制御回路 41 から送信される遊技状態や内部当籤役等の各種データを格納する演出状態格納領域などを有する。発振回路 84 は、サブCPU 81 を動作させるための所定周波数、例えば、22MHz のクロックを生成 (発振) するクロック生成手段を構成する。

【0085】

バックアップRAM 85 は、例えば、電源が不要な書き換え可能な不揮発性メモリであるFRAM (Ferroelectric Random Access Memory) (登録商標) によって構成されている。なお、バックアップRAM 85 は、RTC 86 に電源を供給するバッテリー 87 をSRAM にも接続することにより構成してもよい。

【0086】

バックアップRAM 85 には、RTC 86 が計測する日時を記憶する日時記憶領域と、副制御回路 42 の遊技状態に関する情報、及び演出に関する情報が記憶される遊技情報領域と、エラー情報履歴を格納する格納領域と、ホールメニューで設定するパチスロ 1 の各種設定を格納する格納領域と、を有する。

【0087】

なお、バックアップRAM 85 の遊技情報領域は、任意のタイミング (例えば、副制御回路 42 への電源供給が断られた時 (電断発生時)、スタートコマンドを受信した時など) で、サブRAM 83 の演出状態格納領域をバックアップRAM 85 の遊技情報領域にコピーされ、電源投入処理 (図 19 参照) の直後に、サブRAM 83 の演出状態格納領域にコピーされる。パチスロ 1 は、バックアップRAM 85 の遊技情報領域をサブRAM 83 の演出状態格納領域にコピーすることで、電源投入前に実行されていた遊技の演出を再開することが可能となる。

【0088】

RTC 86 は、パチスロ 1 の電源がオン状態であれば、副制御回路 42 に搭載された電子部品と同じく電源基板 (不図示) から供給される電力で動作し、パチスロ 1 の電源がオフ状態であれば、バッテリー 87 から供給される電力によって動作し、日時を計時する計時手段を構成する。

【0089】

RTC 86 は、サブCPU 81 とI2C (Inter-Integrated Circuit) によって接続され、サブCPU 81 からの要求に応じて計測した日時を表す日時データをサブCPU 81 に送信する。バッテリー 87 は、コイン形リチウム電池などの一次電池、又は、リチウムイオン二次電池などによって構成される。

【0090】

サウンドIC 88 には、スピーカ 20L, 20R が接続されている。サウンドIC 88 は、サブCPU 81 から送信されたサウンドデータにしたがってBGM等の音声信号をスピーカ 20L, 20R から出力させる。

10

20

30

40

50

【0091】

副制御回路42は、第8ランプ群118を駆動するためのドライバIC128a~128cと、第1ランプ群111を駆動するためのドライバIC121a~121bと、第7ランプ群117を駆動するためのドライバIC127a~127dと、ドットマトリクス部119を駆動するためのドライバIC129a~129gとを更に含んで構成される。

【0092】

これらのドライバICとサブCPU81は、シリアルバス通信によって接続されている。サブCPU81にはシリアルバス通信用の通信回路(不図示)が2回線内蔵され、チャンネル0にはドライバIC128a~128c、ドライバIC121a~121b及びドライバIC127a~127dに接続され、チャンネル1にはドライバIC129a~129gが接続されている。

10

【0093】

本実施形態において、第1ランプ群111、第7ランプ群117、第8ランプ群118、ドットマトリクス部119は、発光部を構成する。ドライバIC128a~128c、ドライバIC121a~121b、ドライバIC127a~127d及びドライバIC129a~129g(以下、単に総称して「ドライバIC」ともいう)は、発光駆動手段を構成する。

【0094】

本実施形態において、各ドライバICは、24個の出力端子を有する。したがって、各ドライバICは、チャンネル0からチャンネル23の24系統のLEDを駆動することができる。

20

【0095】

各ドライバICは、出力設定端子を有する。各ドライバICは、出力設定端子の状態(High/Low)に応じて、出力端子の出力を定電流出力とシンク出力とのいずれかに設定する。したがって、各ドライバICの出力端子は、LEDの接続状態に応じて、定電流出力とシンク出力とのいずれかに設定することができる。

【0096】

各ドライバICは、各チャンネルに対応する6ビットのレジスタを内蔵し、レジスタに設定された値に応じたデューティ比の駆動信号を各チャンネルに対して63 μ s周期で出力する。

30

【0097】

例えば、レジスタの値が63の場合には、ドライバICから該当チャンネルにデューティ比が100%(すなわち、パルス幅が63 μ s)の駆動信号が出力され、該当チャンネルのLEDが最も高い輝度で発光する。

【0098】

一方、レジスタの値が0の場合には、ドライバICから該当チャンネルにデューティ比が0%(すなわち、パルス幅が0)の駆動信号が出力され、該当チャンネルのLEDが消灯する。

【0099】

本実施形態では、サブCPU81と各ドライバICとがシリアルバス通信方式により通信を行う。シリアルバス通信は、データ線・クロック線の2線、又は、データ線・クロック線・セレクト線の3線を用いた同期式シリアル通信により、各ドライバICに各ランプ群111, 117, 118及びドットマトリクス部119のLEDの発光態様を制御させるための制御データが送信される。

40

【0100】

サブCPU81は、制御データを出力するための複数の出力ポートを有し、各出力ポートから複数のドライバICに、制御データを送信することができる。本実施形態において、サブCPU81は、制御データを出力するための第1ポートと第2ポートとを有する。第1ポートには、ドライバIC128a~128c、ドライバIC121a~121b及びドライバIC127a~127dが接続されている。第2ポートには、ドライバIC1

50

29a ~ 129g が接続されている。

【0101】

各ドライバICは、複数のアドレス端子を有し、各アドレス端子のレベル(High/Low)によって、サブCPU81の出力ポート内で、ユニークなアドレスが割り当てられている。

【0102】

図8 ~ 図10を参照してそれぞれ説明するように、サブCPU81の各出力ポートから送信される制御データは、各ドライバICに全チャンネルのLEDの輝度を設定する第1モードと、指定したチャンネルのLEDの輝度を設定する第2モードと、全チャンネルのLEDを消灯させるコマンドを含む第3モードとの3つのモードがある。

10

【0103】

なお、図8 ~ 図10において、1行目は、制御データにおけるビット順序を表し、2行目は、制御データに含まれる各パラメータの種別を表し、3行目は、後述するエラー検出データEを表している。

【0104】

エラー検出用データEは、制御データにおける各パラメータの位置によらずに、制御データの所定の位置に挿入される。このため、図8 ~ 図10においては、制御データに含まれる各パラメータの種別と、エラー検出用データEとを2行に分けて示している。すなわち、図8 ~ 図10において、3行目に「E」が示されている各ビットは、エラー検出用データEに相当し、3行目に「E」が示されていない各ビットは、2行目に示すパラメータに相当する。

20

【0105】

図8 ~ 図10において、4行目は、サブCPU81の各出力ポートから実際に送信される制御データを表す。なお、4行目に示す制御データにおいて、「*」は、制御データに応じて設定される値(「0」又は「1」)を表す。

【0106】

(第1モード)

図8に示すように、第1モードにおける制御データは、ヘッダとして、先頭検出ビットHDと、デバイス識別子DEVと、アドレスADRと、モードMDと、所定の位置に挿入されるエラー検出用データE(例えば、固定値「0」)とを含む。

30

【0107】

具体的には、第1モードにおける制御データのヘッダは、22ビットよりなる。先頭検出ビットHDは、1ビット目から9ビット目の9ビットに割り当てられている。先頭検出ビットHDは、図示したように、固定値である。

【0108】

デバイス識別子DEVは、11ビット目から14ビット目の4ビットに割り当てられている。デバイス識別子DEVは、各ドライバICの種別を表す。したがって、同種のドライバICであれば同一な値となる。

【0109】

アドレスADRは、15ビット目から18ビット目及び20ビット目の5ビットに割り当てられている。アドレスADRは、ドライバICを識別するためのアドレスを表す。すなわち、アドレスADRは、制御データの送信先とするドライバICのアドレス端子に設定されたアドレスを表す。

40

【0110】

具体的には、ドライバIC128a ~ 128cのそれぞれにアドレス設定用の5つの端子(不図示)があり、5つの端子それぞれを+5V電源側(PullUp)又は、グランド側(PullDown)に接続すると、ドライバICを識別するためのアドレスが設定される。

【0111】

モードMDは、21ビット目の1ビットに割り当てられている。モードMDは、制御デ

50

ータが第1モードであるか、第2モードであるかを表す。本実施形態においては、モードMDには、第1モードの場合には「0」が設定され、第2モードの場合には「1」が設定される。

【0112】

エラー検出用データEは、10ビット目、19ビット目及び22ビット目の各1ビットに割り当てられている。本実施形態において、エラー検出用データEは、固定値とし、例えば、「0」とする。なお、エラー検出用データEは、固定値以外に、所定の法則に基づく値（例えば、所定周期の循環値）でもよく、パリティビットでもよい。

【0113】

第1モードにおける制御データは、ペイロードとして、チャンネル0からチャンネル23の各輝度データL（図中、Lの添え字は、チャンネル番号を表す）と、エラー検出用データEと、エンドビットENDとを含む。

10

【0114】

チャンネル0からチャンネル23の各輝度データL0～L23は、ペイロードの先頭から6ビット単位で割り当てられている。エラー検出用データEは、ヘッダのエラー検出用データEと同様であり、ペイロードの9ビット目から9ビット周期で挿入されている。エンドビットENDは、制御データの最後のビットを表す。本実施形態において、エンドビットENDは、固定値とし、例えば、「0」とする。

【0115】

このように、第1モードにおける制御データのペイロードは、エラー検出用データEの周期を輝度データLの周期に同期させないことにより、輝度データLのデータパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができるように規定されている。

20

【0116】

第1モードの制御データを受信したドライバICは、割り当てられたアドレスと制御データのアドレスADRとが等しく、予め登録されたデバイス識別子とデバイス識別子DEVとが等しければ、チャンネル0からチャンネル23に対応するレジスタの値を輝度データL0～L23に更新する。

【0117】

（第2モード）

図9に示すように、第2モードにおける制御データのヘッダは、第1モードにおける制御データのヘッダと同様であるため、説明を省略する。

30

【0118】

第2モードにおける制御データは、ペイロードとして、チャンネル数CHNと、チャンネル番号CH（図中、CHの添え字は、aから順次付されている）と、輝度データL（図中、Lの添え字は、aから順次付されている）と、エラー検出用データEと、エンドビットENDとを含む。

【0119】

チャンネル数CHNは、ペイロードの先頭から5ビットに割り当てられている。チャンネル数CHNは、更新対象のチャンネルの数を表す。すなわち、チャンネル数CHNは、以降に続く、チャンネル番号CHと輝度データLとの組の数を表す。

40

【0120】

チャンネル番号CHは、5ビットで表され、チャンネル0からチャンネル23のいずれかのチャンネルの番号を表す。各輝度データLa～Lgは、チャンネル番号CHに続く6ビットで表されている。

【0121】

エラー検出用データEは、第1モードの制御データにおけるエラー検出用データEと同様であり、ペイロードの9ビット目から9ビット周期で挿入されている。エンドビットENDは、第1モードの制御データにおけるエンドビットENDと同様である。

【0122】

50

このように、第2モードにおける制御データのペイロードは、エラー検出用データEの周期を輝度データLの周期に同期させないことにより、輝度データLのデータパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができるように規定されている。

【0123】

第2モードの制御データを受信したドライバICは、割り当てられたアドレスと制御データのアドレスADRとが等しく、予め登録されたデバイス識別子とデバイス識別子DEVとが等しければ、各チャンネル番号CHに対応するレジスタの値を各輝度データLに更新する。

【0124】

(第3モード)

図10に示すように、第3モードにおける制御データは、先頭検出ビットHDと、リセットコマンドRSTと、エラー検出用データEと、エンドビットENDとを含む。

【0125】

先頭検出ビットHDは、第1モードの制御データのヘッダにおける先頭検出ビットHDと同様であり、1ビット目から9ビット目の9ビットに割り当てられている。リセットコマンドRSTは、11ビット目から18ビット目の8ビットに割り当てられている。本実施形態において、リセットコマンドRSTは、図示したように、固定値である。

【0126】

エラー検出用データEは、第1モードの制御データのヘッダにおける先頭検出ビットHDと同様であり、10ビット目の1ビットに割り当てられている。エンドビットENDは、第1モードの制御データにおけるエンドビットENDと同様である。第3モードの制御データを受信したドライバICは、全チャンネルに対応するレジスタの値を0に更新する。

【0127】

このように、チャンネルを個別に制御する場合には、第2モードで制御データを送信することによって、第1モードと比較して制御データのデータ長を短くすることができる。一方、チャンネルを全体的に制御する場合には、チャンネル番号を必要としない第1モードで制御データを送信することによって、第2モードと比較して制御データのデータ長を短くすることができる。

【0128】

また、全てのチャンネルのLEDを消灯させる場合には、第3モードで制御データを送信することによって、第1モード及び第2モードと比較して制御データのデータ長を短くすることができる。

【0129】

以上のように、ドライバICが駆動する発光体の用途に応じて、制御データのデータ長が短くなるモードを選択することにより、サブCPU81とドライバICとの間の伝送負荷を低減させることができ、LEDの応答性を向上させることができる。

【0130】

本実施形態において、ドットマトリクス部119を構成するLEDは、第1発光体群を構成し、各第1ランプ群111、第7ランプ群117、第8ランプ群118を構成するLEDは、第2発光体群を構成する。

【0131】

サブCPU81は、第1発光体群のLEDを駆動するドライバICには、第2ポートから制御データを第2モードで送信する。サブCPU81は、第2発光体群のLEDを駆動するドライバICには、第1ポートから制御データを第1モードで送信する。サブCPU81は、電源投入時や演出の切り替え時などのようにLEDを全消灯させる場合、該当ポートから制御データを第3モードで送信する。

【0132】

なお、本実施形態に係る第2ランプ群112、第3ランプ群113、第4ランプ群114、第5ランプ群115、第6ランプ群116、及び、その他のポートに係る光源部は、

10

20

30

40

50

それぞれ不図示のドライバICを介してサブCPU81によって制御される。

【0133】

<メインROMに記憶されているデータテーブルの構成>

次に、メインROM52に記憶されている各種データテーブルの構成について説明する。

【0134】

[図柄配置表]

まず、図11を参照して、図柄配置表について説明する。図柄配置表は、左リール3L、中リール3C及び右リール3Rのそれぞれの回転方向における各図柄の位置と、各位置に配された図柄の種類を特定するデータとの対応関係を規定する。

10

【0135】

図柄配置表では、リールインデックスが検出されたときに、表示窓4L、4C、4Rの枠内における中段領域に配置される各リールの図柄の位置を「0」と規定する。そして、各リールにおいて、図柄位置「0」を基準としてリールの回転方向(図11における下方向)に進む順に、図柄カウンタに対応する「0」~「20」が、図柄位置として、各図柄に割り当てられる。

【0136】

すなわち、図柄カウンタの値(「0」~「20」)と、図柄配置表とを参照することにより、表示窓4L、4C、4Rの枠内における各リールの上段、中段及び下段の領域に表示されている図柄の種類を特定することができる。例えば、左リール3Lに対応する図柄カウンタの値が「7」であるとき、表示窓4の枠内における左リール3Lの上段、中段及び下段の領域には、それぞれ、図柄位置「8」の「ドン1」、図柄位置「7」の「ベル2」及び図柄位置「6」の「リプレイ」に対応する図柄が表示されている。

20

【0137】

<サブROMに記憶されているデータテーブル>

[月別日テーブル]

図12に示すように、月別日テーブルは、閏年であるときの日数と閏年でないときの日数とが月ごとに対応付けられている。月別日テーブルは、後述する日時更新処理(図26参照)を実行するサブCPU81によって参照される。このように、月別日テーブルを記憶するサブROM82は、月別日テーブル記憶手段を構成する。

30

【0138】

<サブRAMに割り当てられる格納領域>

[日時格納領域]

図13に示すように、日時格納領域は、「年」、「月」、「日」、「曜日」、「時」、「分」及び「秒」をそれぞれ格納するための各領域がサブRAM83に割り当てられる。すなわち、日時格納領域には、日時データが記憶される。

【0139】

以下の説明において、日時格納領域を構成する各領域の値を日時格納領域(年)、日時格納領域(月)、日時格納領域(日)、日時格納領域(曜日)、日時格納領域(時)、日時格納領域(分)、日時格納領域(秒)という。日時格納領域は、後述する日時更新処理(図26参照)を実行するサブCPU81によって使用される。

40

【0140】

<ランプデータ>

本実施形態において、サブROM82には、光の点(増減含む)消灯のパターンに関する演出データであるランプデータ(「発光データ」とも称す)と、複数のパーツデータとが格納されている。このように、サブROM82は、発光データ記憶手段及びパターンデータ記憶手段を構成する。

【0141】

各パーツデータは、パーツデータを識別するための識別情報と、パーツデータの属性を示す属性データと、光の点(増減含む)消灯のパターンを表すパターンデータとを含む。

50

各ランプデータは、パーツデータの識別情報の順序を表す。

【0142】

パターンデータは、再生順に順序付けられた複数の輝度パターンよりなる。各輝度パターンは、制御対象とする各ドライバICの各チャンネルに設定する輝度を表す。各輝度パターンにおいて、ドライバICのチャンネルに設定する輝度は、パターンデータの汎用性が考慮され、8ビットで表されている。

【0143】

なお、ドライバICのチャンネルに設定する輝度は、パターンデータの汎用性が担保されれば、16ビット、32ビット又は64ビットなどで表されていてもよい。パターンデータにおける最後の輝度パターンは、エンドブロックを表す。

【0144】

パーツデータの属性データとしては、ショットと、ショット+チェインと、ループとがある。属性データがショットのパーツデータは、1のランプデータにおいて、一度しか再生されない。

【0145】

ショット+チェインは、連続再生を表す。属性データがショット+チェインのパーツデータは、1のランプデータにおいて、連続再生される。したがって、1のランプデータにおいて、属性データがショット+チェインのパーツデータが連続する場合には、属性データがショット+チェインの一連のパーツデータが繰り返し再生される。

【0146】

すなわち、ランプデータにおける最後の識別情報に対応するパーツデータが再生された後、ランプデータにおける最後の識別情報に対応するパーツデータに含まれる属性データがショット+チェインを表す場合には、ランプデータが表す識別情報の先頭から属性データがショット+チェインを表すパーツデータが検索され、検出されたパーツデータから連続再生が開始される。

【0147】

ループは、繰り返し再生を表す。属性データがループのパーツデータは、1のランプデータにおいて、繰り返し再生される。すなわち、属性データがループのパーツデータの識別情報がランプデータに含まれている場合、このパーツデータが繰り返し再生される。

【0148】

本実施形態におけるランプデータの具体例を説明する。実行する演出に応じたランプデータは、サブCPU81によって選択され、図14に示すサブRAM83の再生状態管理格納領域に読み込まれる。

【0149】

図14において、再生状態管理格納領域には、再生中のパーツデータを識別するためのパーツ番号を格納するエントリパーツ番号格納領域と、パーツデータを識別するための識別情報であるパーツ番号を格納するパーツ番号格納領域と、ランプデータの最後を表すエンドコードを格納するエンドコード格納領域と、ランプデータが表す識別情報のパーツデータに、属性データがショット+チェイン又はループであるパーツデータが含まれているか否かを表すチェイン/ループ設定領域とが含まれる。なお、再生状態管理格納領域に含まれるパーツ番号格納領域の数は、ランプデータが表すパーツデータの数と等しくなる。

【0150】

図15は、再生状態管理格納領域の具体例を示している。図15に示す例において、エントリパーツ番号格納領域には、エントリパーツ番号として「1」（イントロが割り当てられた番号）が格納され、パーツ番号格納領域には、パーツ番号として「1」～「8」がそれぞれ格納され、エンドコード格納領域には、エンドコードが格納され、チェイン/ループ設定領域には、「TRUE」が格納されている。

【0151】

なお、ランプデータが表す識別情報のパーツデータに、属性データがショット+チェイン又はループであるパーツデータが含まれていない場合には、チェイン/ループ設定領域

10

20

30

40

50

には、「FALSE」が格納されている。また、チェーン/ループ設定領域に、「FALSE」が格納されている場合、属性データを検索することなく、パーツデータの繰り返し再生は行われない。(1回限りの再生)

【0152】

図16は、図15に示した再生状態管理格納領域に基づくランプデータの再生例を示す。パーツ番号が「1」のパーツデータは、属性データにショットが設定され、パターンデータに、例えば30秒分のイントロ演出を表す輝度パターンが設定されている。

【0153】

パーツ番号が「2」～「8」の各パーツデータは、属性データにショット+チェーンが設定され、パターンデータに、例えば30秒分の連続演出を表す輝度パターンが設定されている。なお、図示の例では、パーツ番号が「1」～「8」の全てのパーツデータの再生時間が30秒となっているが、パーツデータの長さは、互いに異なってもよい。

【0154】

図中、矢印で示すように、ランプデータの再生が開始されると、パーツ番号が「1」のパーツデータが30秒にわたって再生され、次いで、パーツ番号が「2」～「8」の各パーツデータが30秒にわたって再生される。その際、パーツ番号が「1」から「2」への更新は、再生状態管理格納領域のエントリパーツ番号格納領域の値が、「1」から「2」への更新されることで管理される。

【0155】

エントリパーツ番号格納領域の値を基にエンドコードが検出されると、最後に再生されたパーツ番号が「8」のパーツデータの属性データがショット+チェーンであるため、ランプデータの先頭であるパーツ番号が「1」のパーツデータから、属性データがショット+チェーンであるパーツデータが検索される。

【0156】

図示の例では、パーツ番号が「2」のパーツデータが検出される。したがって、エントリパーツ番号格納領域に「2」がセットされ、図中、矢印で示すように、パーツ番号が「2」のパーツデータからパーツ番号が「2」～「8」の一連のパーツデータが連続して再生される。以降、ランプデータが更新されない限り、パーツ番号が「2」～「8」の一連のパーツデータが繰り返し再生される。

【0157】

ランプデータの再生中にパチスロ1のエラーが検出された場合、サブCPU81は、エラー用のランプデータを再生する。したがって、エラー検出時に再生していたランプデータの再生は、中断する。エラー復帰時には、サブCPU81は、エラー検出時に再生していたランプデータを再生する。

【0158】

このときに、中断したランプデータを最初から再生すると、演出内容が遊技状態にそぐわなくなってしまうことがある。このため、サブROM82には、ランプデータに対応して、中断復帰用ランプデータが格納されている。中断復帰用ランプデータは、通常のランプデータに対して、属性データがショット+チェーンを表していないパーツデータのパーツ番号が除かれている。

【0159】

図17は、中断復帰用ランプデータが格納された再生状態管理格納領域を示している。図17に示す再生状態管理格納領域には、再生状態管理格納領域に「2」が格納され、図15に示した再生状態管理格納領域に格納されたランプデータに対して属性データがショット+チェーンを表していないパーツデータのパーツ番号である「1」(イントロが割り当てられた番号)が除かれている。

【0160】

したがって、図18に示すように、中断復帰用ランプデータは、属性データがショット+チェーンを表す先頭のパーツ番号が「2」のパーツデータから再生が開始される。このように、サブCPU81は、エラー復帰時に中断復帰用ランプデータの再生を開始するこ

10

20

30

40

50

とによって、演出内容が遊技状態にそぐわなくなってしまうこと（演出の途中にも関わらずイントロが再生すること）を防止している。

【0161】

なお、サブCPU81は、後述の属性検索頭出処理（図30のステップS611）で再生状態管理格納領域を使用して、繰り返し再生を行うための頭出しを行っている。図15から図18は、属性検索頭出処理の具体例でもある。

【0162】

なお、サブROM82にランプデータに対応する中断復帰用ランプデータを格納するのに代えて、サブCPU81は、エラー復帰時に、エラー検出時に再生していたランプデータを中断復帰用ランプデータに変換するようにしてもよい。

【0163】

具体的には、サブCPU81は、再生状態管理格納領域に格納されたランプデータに対して属性データがショット+チェインを表していないパーツデータのパーツ番号を除くことにより、中断復帰用ランプデータに変換するようにしてもよい。

【0164】

<主制御回路の動作説明>

次に、図19を参照して、主制御回路41のメインCPU51が、プログラムを用いて実行する各種処理の内容について説明する。

【0165】

[電源投入処理]

まず、メインCPU51の制御で行うパチスロ1の主制御回路41の電源投入処理を、図19を参照しながら説明する。

【0166】

まず、パチスロ1に電源が投入されると、メインCPU51は、電源投入時の初期化処理を行う（S1）。この初期化処理では、バックアップが正常に行われたか、設定変更が適切に行われたか等が判定され、その判定結果に対応した初期化が行われる。

【0167】

次いで、メインCPU51は、一遊技終了時の初期化処理を行う（S2）。この初期化処理では、メインRAM53における指定格納領域のデータをクリアする。なお、ここでいう指定格納領域は、例えば、内部当籤役格納領域や表示役格納領域などの1回の遊技ごとにデータの消去が必要な格納領域である。

【0168】

次いで、メインCPU51は、メダル受付・スタートチェック処理を行う（S3）。この処理では、メダルセンサ35Sやスタートスイッチ16Sの入力のチェック等が行われる。

【0169】

具体的には、メインCPU51は、前回の遊技で再遊技（リプレイ役）に係る表示役が成立したときの自動投入による投入枚数のカウント、メダル投入口13から投入されたメダルによる投入枚数のカウント、及び、BETボタン（MAXBET、1BET）の押下に基づく投入枚数のカウント等が実行され、投入枚数が遊技開始可能枚数に達している場合には、スタートスイッチの状態を判別する。

【0170】

次いで、メインCPU51は、乱数発生器56の第1乱数レジスタ（不図示）から乱数値（0～65535）を抽出し、該抽出した乱数値をメインRAM53に設けられた乱数値格納領域（不図示）に格納する（S4）。次いで、メインCPU51は、乱数発生器56の第2乱数レジスタ（不図示）からリアル演出及びロックの制御で用いる演出用乱数値を抽出し、該抽出した演出用乱数値をメインRAM53に設けられた演出用乱数値格納領域（不図示）に格納する（S5）。なお、本実施形態では、演出用乱数値は、0～127の範囲から抽出される。

【0171】

10

20

30

40

50

そして、抽出した各種乱数値が所定の乱数値格納領域に格納されると、メインCPU 51は、内部抽籤処理を行う(S 6)。この処理では、S 4で抽出した乱数値に基づいた抽籤により内部当籤役の決定が行われる。

【0172】

次いで、メインCPU 51は、遊技ロック抽籤処理を行う(S 7)。遊技ロック抽籤処理において、メインCPU 51は、遊技状態に応じた遊技ロック抽籤テーブル(不図示)を参照し、演出用乱数値に基づいて、遊技ロック抽籤を行う。

【0173】

次いで、メインCPU 51は、リール停止初期設定処理を行う(S 8)。リール停止初期設定処理において、メインCPU 51は、内部当籤役に基づいて、リール停止初期設定テーブル(不図示)からルールの停止制御に係る各種情報を取得する。

10

【0174】

次いで、メインCPU 51は、スタートコマンド送信処理を行う(S 9)。具体的には、メインCPU 51は、スタートコマンドを副制御回路42に送信する。なお、スタートコマンドは、内部当籤役等を特定するパラメータ、遊技ロックの種別及びロック時間等を含んで構成される。

【0175】

次いで、メインCPU 51は、ウェイト処理を行う(S 10)。この処理では、メインCPU 51は、前回の遊技開始から所定時間(例えば、4.1秒)を経過していない場合、該所定時間が経過するまで待ち時間を消化する。

20

【0176】

次いで、メインCPU 51は、リール回転開始処理を行う(S 11)。この処理において、メインCPU 51は、全リールの回転開始を要求する。リール回転開始処理において、メインCPU 51は、全リールの回転開始を要求する。全リールの回転開始が要求されると、一定の周期(1.1172 msec)で実行される後述の割込処理(図20参照)により、3つのステッピングモータ61L、61C、61Rの駆動が制御され、左リール3L、中リール3C及び右リール3Rの回転が開始される。

【0177】

次いで、メインCPU 51は、引込優先順位格納処理を行う(S 12)。この処理では、メインCPU 51は、引込優先順位データを取得して、引込優先順位データ格納領域(不図示)に格納する。

30

【0178】

次いで、メインCPU 51は、リール停止制御処理を行う(S 13)。この処理では、左ストップボタン17L、中ストップボタン17C及び右ストップボタン17Rがそれぞれ押されたタイミングと内部当籤役とに基づいて該当するリールの回転が停止される。すなわち、本実施形態では、リール停止制御処理を実行する主制御回路41は停止制御手段を構成する。

【0179】

次いで、メインCPU 51は、遊技ロック処理を行う(S 14)。この処理において、メインCPU 51は、遊技の進行を無効化する又は遅延させる。メインCPU 51は、決定された遊技ロック種別を実行する期間に相当する値をロックタイマにセットし、ロックタイマの値が「0」になるまで待機する。その間、メインCPU 51は、遊技者のあらゆる操作を受け付けない。

40

【0180】

次いで、メインCPU 51は、入賞検索処理を行う(S 15)。この処理では、メインCPU 51は、図柄コード格納領域(不図示)のデータを表示役格納領域(不図示)に格納する。また、この処理では、左リール3L、中リール3C及び右リール3Rが全て停止した後有効ライン(入賞判定ライン)に表示された図柄の組合せと、図柄組合せテーブル(不図示)とを照合する。そして、メインCPU 51は、有効ラインに表示役が表示されたか否かを判定し、その判定結果を表示役格納領域に格納するようにしてもよい。

50

【0181】

次いで、メインCPU51は、メダル払出処理を行う(S16)。メダル払出処理は、本発明に係る遊技媒体付与手段の一具体例を示す。この処理では、S15において決定された表示役の払出枚数に基づいて、ホッパー33の駆動やクレジット枚数の更新が行われ、メダルの払い出しが行われる。この際、本実施形態では、図柄組合せテーブル(不図示)に示すように、メダルの投入枚数が1~3枚であり、メダルの払出枚数は表示役に応じて異なるが、その最大払出枚数(払出上限)は15枚である。

【0182】

次いで、メインCPU51は、RT制御処理を行う(S17)。この処理では、メインCPU51は、RT遊技状態を管理する。メインCPU51は、RT遷移テーブル(不図示)を参照し、移行元(現在)のRT遊技状態において成立し得るRT遊技状態の移行条件をチェックし、RT遊技状態の移行条件が成立していると判別したとき、RT遷移テーブル(不図示)を参照し、移行条件に基づいて、移行先のRT遊技状態に移行する。

10

【0183】

次いで、メインCPU51は、払出終了コマンド送信処理を行う(S18)。具体的には、メインCPU51は、払出終了コマンドを副制御回路42に送信する。

【0184】

次いで、メインCPU51は、ボーナス終了チェック処理を行う(S19)。この処理では、メインCPU51は、ボーナスゲームの終了契機を管理するための各種カウンタを参照して、ボーナスゲームの作動を終了するか否かをチェックする。

20

【0185】

次いで、メインCPU51は、ボーナス作動チェック処理を行う(S20)。この処理では、メインCPU51は、ボーナスゲームの作動を開始するか否か、及び、再遊技を行うか否かをチェックする。ボーナス作動チェック処理が終了すると、メインCPU51は、処理をS2に戻し、S2以降の処理を繰り返す。

【0186】

[メインCPUの制御による割込処理(1.1172msec)]

次に、図20を参照して、メインCPU51に内蔵されたタイマ(不図示)の制御による定周期(1.1172msec毎)に行われる割込処理について説明する。

【0187】

まず、メインCPU51は、レジスタの退避を行う(S351)。次いで、メインCPU51は、入力ポートチェック処理を行う(S352)。この処理では、ストップスイッチ17S等の各種スイッチから入力される信号がチェックされる。

30

【0188】

次いで、メインCPU51は、タイマ更新処理を行う(S353)。この処理では、メインCPU51は、例えば、割込処理毎にロックタイマの値を減算する処理を行う。次いで、メインCPU51は、通信データ送信処置を行う(S354)。この処理では、主に、各種コマンドを主制御回路41及び副制御回路42に適宜送信する。

【0189】

次いで、メインCPU51は、リール制御処理を行う(S355)。この処理では、メインCPU51は、全リールの回転開始が要求されたときに、左リール3L、中リール3C及び右リール3Rの回転を開始し、その後、各リールが一定速度で回転するように、3つのステッピングモータ61L, 61C, 61Rを駆動制御する。また、滑り駒数が決定されたときは、メインCPU51は、該当するリールの図柄カウンタを滑り駒数分だけ更新する。そして、メインCPU51は、更新された図柄カウンタが停止予定位置に対応する値に一致する(停止予定位置の図柄が表示窓の有効ライン(入賞判定ライン)上の領域に到達する)のを待って、該当するリールの回転の減速及び停止が行われるように、対応するステッピングモータを駆動制御する。また、本実施形態では、S355の処理において、前述した通常の加速処理、定速処理及び停止処理だけでなく、加速処理時にリール演出パターンが設定されている場合には、該リール演出パターンに対応するリール演出(リ

40

50

ールアクション)及びロックの制御処理も行う。

【0190】

次いで、メインCPU51は、ランプ・7セグ駆動処理を行う(S356)。この処理では、メインCPU51は、7セグ表示器6を駆動制御して、払出枚数やクレジット枚数などを表示する。次いで、メインCPU51は、レジスタの復帰処理を行う(S357)。そして、その後、メインCPU51は、割込処理を終了する。

【0191】

<副制御回路の動作説明>

次に、図21～図32を参照して、副制御回路42のサブCPU81が、プログラムを用いて実行する各種処理(タスク)の内容について説明する。

【0192】

[電源投入処理]

図21は、電源投入時に実行される副制御回路42の電源投入処理を示すフローチャートである。

【0193】

まず、サブCPU81は、初期化処理を実行する(S361)。初期化処理において、サブCPU81は、サブRAM83等のエラーチェックに加えて、各種ドライバの初期化を行う。

【0194】

S361で初期化されるタスクは、コマンド受信割込同期のタスクグループである主基板通信タスク(図22参照)、並びに、タイマ割込同期のタスクグループであるサウンド制御タスク(図27参照)及びランプ制御タスク(図30参照)等を含む。

【0195】

次に、サブCPU81は、ランプ制御タスクを起動する(S362)。ランプ制御タスクでは、サブCPU81は、2ms毎に送信されるタイマ割込イベントメッセージが受信されるのを待ち、タイマ割込イベントメッセージを受信したことに応じて、LED群21などの各種ランプの点灯状態を制御する。このように、ランプ制御タスクを実行するサブCPU81は、制御手段を構成する。

【0196】

次に、サブCPU81は、サウンド制御タスクを起動する(S363)。サウンド制御タスクでは、サブCPU81は、ランプ制御タスクと同様に、スピーカ20L、20Rなどの各種スピーカの出音状態を制御する。

【0197】

次に、サブCPU81は、主基板通信タスクを起動する(S364)。主基板通信タスクでは、サブCPU81は、主制御回路41から送信されるコマンドに対する受信及び解析と解析したコマンドに基づいて演出を決定(抽籤)する。

【0198】

次に、サブCPU81は、RTC86から日時データを取得し、サブRAM83に割り当てられた日時格納領域(図13参照)に格納する(S365)。このように、サブCPU81は、日時取得手段を構成する。

【0199】

次に、サブCPU81は、サブCPU81に備えられたクロックカウンタ(不図示)からカウント値を取得しクロックカウンタPとしてサブRAM83に格納する(S366)。S366の処理を実行した後、サブCPU81は、電源投入処理を終了する。なお、クロックカウンタとは、サブCPU81に内蔵され発振回路84(図7参照)が発振する所定周波数のクロックをカウントするカウンタ回路であり、発振回路84のクロックをカウントするクロックカウンタを備えるサブCPU81は、カウント手段を構成する。

【0200】

[主基板通信タスク]

次に、図22を参照して、サブCPU81により行われる主基板通信タスクについて説

10

20

30

40

50

明する。

【0201】

まず、サブCPU81は、主制御回路41から送信されたコマンドの受信チェックを行う(S501)。次いで、サブCPU81は、受信チェックの結果に基づいて受信したコマンドが有効であるか否かを判別する(S502)。

【0202】

例えば、サブCPU81は、受信したコマンドのデータ長が8Byteである第1条件と、受信したコマンドがスタートコマンド、リール停止コマンド、表示コマンド、払出終了コマンド、ボーナス開始コマンド、ボーナス終了コマンド及び無操作コマンドなど予め登録されたコマンドである第2条件と、受信したコマンドの8Byte目に格納されたサム値が正しい第3条件との3つの条件が成立した場合には、受信チェックのチェック結果に受信したコマンドが有効であると反映され、いずれかの条件が成立しなかった場合には、受信チェックのチェック結果に受信したコマンドが有効でないと反映される。

10

【0203】

S502において、サブCPU81が、受信したコマンドが有効でないと判別したとき(S502がNO判定の場合)、サブCPU81は、処理をS501に戻し、S501以降の処理を繰り返す。

【0204】

一方、S502において、サブCPU81が、受信したコマンドが有効であると判別したとき(S502がYES判定の場合)、サブCPU81は、受信したコマンドに基づいて、メッセージキューにメッセージを格納する(S503)。なお、メッセージキューとは、プロセス間で情報を交換するための機構である。そして、S503の処理後、サブCPU81は、処理をS501に戻し、S501以降の処理を繰り返す。

20

【0205】

[演出登録タスク]

次に、図23を参照して、サブCPU81により行われる演出登録タスクについて説明する。

【0206】

まず、サブCPU81は、メッセージキューからメッセージを取り出す(S511)。次いで、サブCPU81は、メッセージキューにメッセージが有るか否かを判別する(S512)。S512において、サブCPU81が、メッセージキューにメッセージが無いと判別したとき(S512がNO判定のとき)、サブCPU81は、後述のS515の処理を行う。

30

【0207】

一方、S512において、サブCPU81が、メッセージキューにメッセージが有ると判別したとき(S512がYES判定のとき)、サブCPU81は、メッセージから遊技情報を複写する(S513)。この処理では、例えば、パラメータによって特定される、内部当籤役、回転が停止したリールの種別、表示役、遊技状態フラグ等の各種データがサブRAM83に設けられた格納領域(不図示)に複写される。

40

【0208】

次いで、サブCPU81は、演出内容決定処理を行う(S514)。この処理では、サブCPU81は、受信したコマンドの種別に応じて、演出内容の決定や演出データの登録等を行う。なお、演出内容決定処理の詳細については、後述の図24を参照しながら後で説明する。

【0209】

次いで、サブCPU81は、サウンドデータの登録を行う(S515)。次いで、サブCPU81は、ランプデータの登録を行う(S516)。なお、これらの登録処理は、S514の演出内容決定処理において登録された演出データに基づいて行われる。また、本実施形態において、サブCPU81は、サウンドデータのリクエスト番号が決定されていた場合に限りサウンドデータのリクエスト番号をサブRAM83の登録領域に記憶するこ

50

とでサウンドデータを登録し、ランプデータのリクエスト番号が決定されていた場合に限りランプデータのリクエスト番号をサブRAM 83の登録領域に記憶することでランプデータを登録する。S 5 1 6の後、サブCPU 81は、処理をS 5 1 1に戻し、S 5 1 1以降の処理を繰り返す。

【0210】

[演出内容決定処理]

次に、図24を参照して、演出登録タスクのフローチャート(図23参照)中のS 5 1 4で行う演出内容決定処理について説明する。

【0211】

まず、サブCPU 81は、スタートコマンド受信時であるか否かを判別する(S 5 2 1)。

10

【0212】

S 5 2 1において、サブCPU 81が、スタートコマンド受信時であると判別したとき(S 5 2 1がYES判定の場合)、サブCPU 81は、スタートコマンド受信時処理を行う(S 5 2 2)。この処理では、サブCPU 81は、演出用乱数値を抽出し、内部当籤役等に基づいて演出番号を抽籤により決定して登録する。ここで、演出番号は、今回実行する演出内容を指定するデータである。

【0213】

次いで、サブCPU 81は、登録されている演出番号に応じて、スタート時の演出データを登録する(S 5 2 3)。演出データは、アニメーションデータ、サウンドデータ及びランプデータを指定するデータである。それゆえ、演出データが登録されると、対応するアニメーションデータ等が決定され、表示装置による表示等の演出が実行される。そして、S 5 2 3の処理後、サブCPU 81は、演出内容決定処理を終了し、処理を演出登録タスク(図23参照)のS 5 1 5に移す。

20

【0214】

一方、S 5 2 1において、サブCPU 81が、スタートコマンド受信時でないとは判別したとき(S 5 2 1がNO判定の場合)、サブCPU 81は、リール停止コマンド受信時であるか否かを判別する(S 5 2 4)。

【0215】

S 5 2 4において、サブCPU 81が、リール停止コマンド受信時であると判別したとき(S 5 2 4がYES判定の場合)、サブCPU 81は、登録されている演出番号及び作動ストップボタンの種別に応じて、停止時の演出データを選択する(S 5 2 5)。その後、サブCPU 81は、演出内容決定処理を終了し、処理を演出登録タスク(図23参照)のS 5 1 5に移す。

30

【0216】

一方、S 5 2 4において、サブCPU 81が、リール停止コマンド受信時でないとは判別したとき(S 5 2 4がNO判定の場合)、サブCPU 81は、表示コマンド受信時であるか否かを判別する(S 5 2 6)。

【0217】

S 5 2 6において、サブCPU 81が、表示コマンド受信時であると判別したとき(S 5 2 6がYES判定の場合)、サブCPU 81は、表示コマンド受信時処理を行う(S 5 2 7)。この処理では、サブCPU 81は、表示役等に基づいて演出番号を抽籤により決定して登録する。ここで、演出番号は、今回実行する演出内容を指定するデータである。その後、サブCPU 81は、演出内容決定処理を終了し、処理を演出登録タスク(図23参照)のS 5 1 5に移す。

40

【0218】

一方、S 5 2 6において、表示コマンド受信時でないとは判別したとき(S 5 2 6がNO判定の場合)、サブCPU 81は、払出終了コマンド受信時であるか否かを判別する(S 5 2 8)。S 5 2 8において、サブCPU 81は、払出終了コマンド受信時であると判別したとき(S 5 2 8がYES判定の場合)、サブCPU 81は、払出終了コマンド受信時

50

処理を行う（S529）。その後、サブCPU81は、演出内容決定処理を終了し、処理を演出登録タスク（図23参照）のS515に移す。

【0219】

一方、S528において、サブCPU81が、払出終了コマンド受信時でないとは判断したとき（S528がNO判定の場合）、サブCPU81は、ボーナス開始コマンド受信時であるか否かを判断する（S530）。

【0220】

S530において、サブCPU81が、ボーナス開始コマンド受信時であると判断したとき（S530がYES判定の場合）、サブCPU81は、ボーナス開始用の演出データを登録する（S531）。その後、サブCPU81は、演出内容決定処理を終了し、処理を演出登録タスク（図23参照）のS515に移す。

10

【0221】

一方、S530において、サブCPU81が、ボーナス開始コマンド受信時でないとは判断したとき（S530がNO判定の場合）、サブCPU81は、ボーナス終了コマンド受信時であるか否かを判断する（S532）。S532において、サブCPU81が、ボーナス終了コマンド受信時であると判断したとき（S532がYES判定の場合）、サブCPU81は、ボーナス終了用の演出データを登録する（S533）。その後、サブCPU81は、演出内容決定処理を終了し、処理を演出登録タスク（図23参照）のS515に移す。

【0222】

S532において、サブCPU81が、ボーナス終了コマンド受信時ではないとは判断したとき（S532がNO判定の場合）、サブCPU81は、無操作コマンド受信時であるか否かを判断する（S534）。S534において、サブCPU81が、無操作コマンド受信時ではないとは判断したとき（S534がNO判定の場合）、サブCPU81は、演出内容決定処理を終了し、処理を演出登録タスク（図23参照）のS515に移す。

20

【0223】

一方、S534において、サブCPU81が、無操作コマンド受信時であると判断したとき（S534がYES判定の場合）、サブCPU81は、無操作コマンド受信時処理を行う（S535）。なお、無操作コマンド受信時処理の詳細については、後述の図25を参照しながら後で説明する。S535の処理後、サブCPU81は、演出内容決定処理を終了し、処理を演出登録タスク（図23参照）のS515に移す。

30

【0224】

[無操作コマンド受信時処理]

次に、図25を参照して、演出内容決定処理のフローチャート（図24参照）中のS535で行う無操作コマンド受信時処理について説明する。

【0225】

無操作コマンド受信時処理において、サブCPU81は、図26に示す日時更新処理を実行する（S540）。S540の処理を実行した後、サブCPU81は、無操作コマンド受信時処理を終了する。

【0226】

無操作コマンドは、約10msec周期で、主制御回路41から送信される。図26に示す日時更新処理は、無操作コマンド受信時処理に限らず、周期的に実行される他の処理やタスクから実行されるようにしてもよい。

40

【0227】

[日時更新処理]

次に、図26を参照して、日時更新処理について説明する。

【0228】

まず、サブCPU81は、クロックカウンタからカウント値を取得しクロックカウンタCとしてサブRAM83に格納する（S541）。次に、サブCPU81は、クロックカウンタCからクロックカウンタPを減じたクロックカウンタDを算出する（S542）。

50

このように、サブCPU81は、経過時間算出手段を構成する。

【0229】

次に、サブCPU81は、クロックカウンタDが1秒に相当するクロックカウンタ値（以下、単に「1秒カウンタ」という）以上であるか否かを判断する（S543）。本実施形態において、サブCPU81は、22MHzのクロックで動作しているため、1秒カウンタは、22,000,000（パルス）となる。

【0230】

S543において、クロックカウンタDが1秒カウンタ以上でないと判断した場合には（NO）、サブCPU81は、日時更新処理を終了する。S543において、クロックカウンタDが1秒カウンタ以上であると判断した場合には（YES）、サブCPU81は、クロックカウンタDから1秒カウンタを減算する（S544）。 10

【0231】

S544の処理を実行した後、サブCPU81は、クロックカウンタPに1秒カウンタを加算する（S545）。次に、サブCPU81は、日時格納領域（秒）に1を加算する（S546）。

【0232】

次に、サブCPU81は、日時格納領域（秒）が60以上であるか否かを判断する（S547）。S547において、日時格納領域（秒）が60以上でないと判断した場合には（NO）、サブCPU81は、S543の処理を実行する。

【0233】

S547において、日時格納領域（秒）が60以上であると判断した場合には（YES）、サブCPU81は、日時格納領域（秒）を0に更新し、日時格納領域（分）に1を加算する（S548）。 20

【0234】

S548の処理を実行した後、サブCPU81は、日時格納領域（分）が60以上であるか否かを判断する（S549）。S549において、日時格納領域（分）が60以上でないと判断した場合には（NO）、サブCPU81は、S543の処理を実行する。

【0235】

S549において、日時格納領域（分）が60以上であると判断した場合には（YES）、サブCPU81は、日時格納領域（分）を0に更新し、日時格納領域（時）に1を加算する（S550）。 30

【0236】

S550の処理を実行した後、サブCPU81は、日時格納領域（時）が24以上であるか否かを判断する（S551）。S551において、日時格納領域（時）が24以上でないと判断した場合には（NO）、サブCPU81は、S543の処理を実行する。

【0237】

S551において、日時格納領域（時）が24以上であると判断した場合には（YES）、サブCPU81は、日時格納領域（時）を0に更新し、日時格納領域（日）及び日時格納領域（曜日）に1を加算する（S552）。 40

【0238】

S552の処理を実行した後、サブCPU81は、日時格納領域（曜日）が7以上であるか否かを判断する（S553）。S553において、日時格納領域（曜日）が7以上であると判断した場合には（YES）、サブCPU81は、日時格納領域（曜日）を0に更新する（S554）。

【0239】

S553において、日時格納領域（曜日）が7以上でないと判断した場合（NO）、又は、S554の処理を実行した後、サブCPU81は、日時格納領域（年）が表す年が閏年であるか否かを算出する（S555）。

【0240】

なお、閏年の算出は、西暦年を「4」、「100」及び「400」で除算して余りがな 50

い年（例えば、2000年等）、及び、西暦年を「4」を除算して余りがない、且つ、西暦年を「100」で除算して余りがある年（例えば、2020年等）が閏年となる。

【0241】

次に、サブCPU81は、月別日テーブル（図12参照）によって、S555の算出結果と、日時格納領域（月）とに対応付けられている日数（以下、「月日数」ともいう）を特定し、日時格納領域（日）が月日数以上であるか否かを判断する（S556）。

【0242】

S556において、日時格納領域（日）が月日数以上でないと判断した場合には（NO）、サブCPU81は、S543の処理を実行する。S556において、日時格納領域（日）が月日数以上であると判断した場合には（YES）、サブCPU81は、日時格納領域（日）を1に更新し、日時格納領域（月）に1を加算する（S557）。

10

【0243】

S557の処理を実行した後、サブCPU81は、日時格納領域（月）が12以上であるか否かを判断する（S558）。S558において、日時格納領域（月）が12以上でないと判断した場合には（NO）、サブCPU81は、S543の処理を実行する。

【0244】

S558において、日時格納領域（月）が12以上であると判断した場合には（YES）、サブCPU81は、日時格納領域（月）を1に更新し、日時格納領域（年）に1を加算する（S559）。S559の処理を実行した後、サブCPU81は、S543の処理を実行する。

20

【0245】

このように、日時更新処理を実行するサブCPU81は、経過時間算出手段を構成する。日時更新処理は、日時格納領域に記憶された日時データに規定時間として1秒を加算していくことにより、日時データを更新する。したがって、本来であれば、RTC86から日時データを取得し、取得した日時データを日時格納領域に記憶することで、日時格納領域の日時データを更新するが、日時更新処理は、RTC86とシリアル通信を行うことなく日時格納領域の日時データを更新することにより、サブCPU81はRTC86との通信に係る処理負荷を省略することができる。

【0246】

一般的に、クロックカウントから経過日時を算出する場合には、クロックカウントを1日相当のクロックカウントで除算し、その商から、経過年数、経過月数及び経過日数を算出し、その余りを1時間相当のクロックカウントで除算して経過時間を算出し、その余りを1分相当のクロックカウントで除算して経過分数を算出し、その余りを1秒相当のクロックカウントで除算して経過秒数を算出する。

30

【0247】

CPUによる除算処理は、浮動小数点演算又は繰り返し演算によって実行される。このため、除算は、四則演算のなかで、CPUにかかる処理負荷が最も高い。本実施形態における日時更新処理は、処理負荷が高い除算処理を行わずに加算処理だけで日時格納領域に記憶された日時データを更新することにより、サブCPU81にかかる負荷を低減させている。

40

【0248】

また、本実施形態では、電源投入処理（図21）でRTC86から日時データを取得し、クロックカウンタを用いて日時データを更新する説明をしたが、本発明はこれに限定されない、例えば、ステップS550の代わりに1時間毎にRTC86から日時データを読み込むようにしてもよい。

【0249】

[サウンド制御タスク]

次に、図27を参照して、サブCPU81により行われるサウンド制御タスクについて説明する。

【0250】

50

まず、サブCPU 81は、サブROM 82に記憶されている制御プログラムのうち所定のプログラムをサブRAM 83に転送するサウンド関数RAM転送処理を実行する(S 581)。サウンド関数RAM転送処理については、図28を参照して後述する。

【0251】

次に、サブCPU 81は、演出登録タスク(図23のS 515参照)で登録されたサウンドデータが更新されたか否かを判断する(S 582)。このサウンドデータは、一定の期間におけるスピーカ20L, 20R(図2参照)を駆動制御するためのデータである。

【0252】

S 582において、サブCPU 81は、登録されたサウンドデータが更新されたと判断した場合には(Y E S)、S 583の処理を実行し、登録されたサウンドデータが更新されなかったと判断した場合には(N O)、S 584の処理を実行する。

10

【0253】

S 583において、サブCPU 81は、登録されたサウンドデータを取得する(S 583)。S 583の処理を実行した後、サブCPU 81は、S 584の処理を実行する。S 584において、サブCPU 81は、サウンドデータとサウンドデータの再生位置とに応じてサウンド制御データを生成するサウンド制御データ生成処理を実行する。

【0254】

サウンド制御データは、所定の周期毎のスピーカ20L, 20Rを駆動制御するためのデータである。すなわち、サブCPU 81は、一定の期間におけるスピーカ20L, 20Rを駆動制御するためのデータであるサウンドデータを、所定の周期毎に分けてサウンド制御データを作成する。

20

【0255】

S 584の処理を実行した後、サブCPU 81は、S 585の処理を実行する。S 585において、サブCPU 81は、サウンド制御データ生成処理(S 584)で生成したサウンド制御データをサウンドドライバ(ソフトウェア)に登録する。

【0256】

このように、サウンド制御データをサウンドドライバに登録することによって、サウンドドライバの処理を実行するサブCPU 81は、サウンド制御データをサウンドIC 88に送信する。また、サウンド制御データをサウンドIC 88に送信する際に、サブCPU 81は、サウンドICの再生状態等を受信する。

30

【0257】

S 585の処理を実行した後、サブCPU 81は、S 586の処理を実行する。S 586において、サブCPU 81は、再生中のサウンドが再生終了である否かを判断する。S 586において、サブCPU 81は、再生中のサウンドが再生終了であると判断した場合には(Y E S)、S 587の処理を実行し、再生中のサウンドが再生終了でないと判断した場合には(N O)、S 582の処理を実行する。

【0258】

S 587において、サブCPU 81は、再生中のサウンドがループ再生である否かを判断する。S 587において、サブCPU 81は、再生中のサウンドがループ再生であると判断した場合には(Y E S)、S 588の処理を実行し、再生中のサウンドがループ再生でないと判断した場合には(N O)、S 582の処理を実行する。

40

【0259】

S 588において、サブCPU 81は、サウンドデータ頭出処理を実行する。サウンドデータ頭出処理において、サブCPU 81は、サウンドデータの再生位置を先頭に戻す。S 588の処理を実行した後、サブCPU 81は、S 582の処理を実行する。

【0260】

[サウンド関数RAM転送処理]

次に、図28を参照して、サウンド関数RAM転送処理について説明する。

【0261】

50

まず、サブCPU 81は、サウンド制御タスクにおけるサウンド制御データ生成処理（図27のS584参照）を実行するためのプログラムをサブROM 82からサブRAM 83に転送する（S591）。

【0262】

次に、サブCPU 81は、サウンドドライバをサブROM 82からサブRAM 83に転送する（S592）。S592の処理を実行した後、サブCPU 81は、サウンド関数RAM転送処理を終了する。

【0263】

サブCPU 81は、上位プログラムであるサウンド制御タスクのプログラムからサブRAM 83に転送したサウンド制御データ生成処理のプログラムを実行する。また、サブCPU 81は、サウンドドライバとしての処理をサブRAM 83に転送したプログラムにしたがって実行する。

10

【0264】

このように、サブCPU 81は、サウンド関数RAM転送処理において、相対的に処理負荷が高い所定のプログラムであるサウンドデータ生成処理を実行するためのプログラム及びサウンドドライバをサブROM 82からサブRAM 83に転送して実行することによって、処理速度を向上させている。

【0265】

[サウンド制御タスク及びサウンド関数RAM転送処理の具体例]

図29を参照して、図27及び図28を参照してそれぞれ説明したサウンド制御タスク及びサウンド関数RAM転送処理について具体的に説明する。なお、図29において、サウンド制御タスク及びサウンド関数RAM転送処理は、それぞれC言語で表されている。

20

【0266】

「SoundRamTrans()」は、図28に示したサウンド関数RAM転送処理を実行するためのプログラムである。サブROM 82のアドレス「_SoundMakeData」で表される位置には、サウンド制御タスクのステップS584の処理を実行するためのプログラムが格納されている。

【0267】

サウンド関数RAM転送処理のステップS591では、メモリコピー関数（memcpy）によって、サブROM 82のアドレス「_SoundMakeData」からサブRAM 83のアドレス「__SoundMakeData」で表される位置に、プログラムサイズ（sizeof(_SoundMakeData)）分のメモリコピーが行われる。

30

【0268】

このように、サウンド制御タスクのステップS584の処理を実行するためのプログラムがサブRAM 83に転送される。なお、本実施形態において、サブRAM 83のアドレス「__SoundMakeData」で表される位置には、当該プログラムを格納するための領域がサウンド関数RAM転送処理の実行前に確保されていることとする。

【0269】

サブROM 82のアドレス「_SetSoundDriver」で表される位置には、サウンド制御タスクのステップS585の処理を実行するためのプログラムが格納されている。サウンド関数RAM転送処理のステップS592では、メモリコピー関数（memcpy）によって、サブROM 82のアドレス「_SetSoundDriver」からサブRAM 83のアドレス「__SetSoundDriver」で表される位置に、プログラムサイズ（sizeof(_SetSoundDriver)）分のメモリコピーが行われる。

40

【0270】

このように、サウンド制御タスクのステップS585の処理を実行するためのプログラムがサブRAM 83に転送される。なお、本実施形態において、サブRAM 83のアドレス「__SetSoundDriver」で表される位置には、当該プログラムを格納するための領域がサウンド関数RAM転送処理の実行前に確保されていることとする。

【0271】

50

「SoundTask()」は、図 2 7 に示したサウンド制御タスクを実行するためのプログラムである。なお、図 2 9 に示した「SoundTask()」は、ステップ S 5 8 3 ~ S 5 8 5 の処理を表し、他の処理の記載は省略されている。

【 0 2 7 2 】

「SoundTask()」において、「index」は、サウンドデータを識別するためのリクエスト番号を表す変数として用いられ、「SoundData」は、サブ R O M 8 2 のサウンドデータの格納先のアドレスを表すポインタ変数として用いられ、「SoundBuf」は、サブ R A M 8 3 のサウンド制御データの格納先のアドレスを表すポインタ変数として用いられる。

【 0 2 7 3 】

「GetSoundData()」は、サウンド制御タスクのステップ S 5 8 3 の処理を実行し、登録されたサウンドデータを取得する。具体的には、「GetSoundData()」は、「index」によって登録されたサウンドデータのリクエスト番号が指定されると、指定されたサウンドデータの格納先のアドレスを「SoundData」に格納する。

10

【 0 2 7 4 】

「SoundMakeData()」は、図 2 8 に示したサウンド関数 R A M 転送処理のステップ S 5 9 1 でサブ R A M 8 3 のアドレス「__SoundMakeData」に転送されたプログラム「__SoundMakeData()」のアクセス関数である。

【 0 2 7 5 】

「__SoundMakeData()」は、引数「dat」が表すサウンドデータの格納先のアドレスが指定されると、アドレスが指定されたサウンドデータに基づいてサウンド制御データをサブ R A M 8 3 に生成し、生成したサウンド制御データの格納先のアドレスを引数「buf」に格納する。

20

【 0 2 7 6 】

したがって、「SoundTask()」における「SoundMakeData()」は、サウンド制御タスクのステップ S 5 8 4 の処理を実行し、サウンドデータの格納先のアドレスを表すポインタ変数「SoundData」が指定されることで、ポインタ変数「SoundBuf」にサウンド制御データの格納先であるサブ R A M 8 3 のアドレスを格納する。

【 0 2 7 7 】

「SetSoundDriver()」は、図 2 8 に示したサウンド関数 R A M 転送処理のステップ S 5 9 1 でサブ R A M 8 3 のアドレス「__SetSoundDriver」に転送されたプログラム「__SetSoundDriver()」のアクセス関数である。「__SetSoundDriver()」は、引数「buf」が表すサウンド制御データの格納先であるサブ R A M 8 3 のアドレスが指定されると、指定されたアドレスをサウンドドライバに登録する。

30

【 0 2 7 8 】

したがって、「SoundTask()」における「SetSoundDriver()」は、サウンド制御タスクのステップ S 5 8 5 の処理を実行し、サウンド制御データの格納先のアドレスを表す「SoundBuf」が指定されることで、サウンド制御データの格納先であるサブ R A M 8 3 のアドレスをサウンドドライバに登録する。

【 0 2 7 9 】

[ランプ制御タスク]

40

次に、図 3 0 を参照して、サブ C P U 8 1 により行われるランプ制御タスクについて説明する。

【 0 2 8 0 】

まず、サブ C P U 8 1 は、サブ R O M 8 2 に記憶されている制御プログラムのうち所定のプログラムをサブ R A M 8 3 に転送するランプ関数 R A M 転送処理を実行する (S 6 0 1)。ランプ関数 R A M 転送処理については、図 3 2 を参照して後述する。

【 0 2 8 1 】

次に、サブ C P U 8 1 は、演出登録タスク (図 2 3 の S 5 1 6 参照) 等で登録されたランプデータをサブ R A M 8 3 に読み込むランプデータ読み込み処理を実行する (S 6 0 2)。ランプデータ読み込み処理については、図 3 1 を参照して後述する。

50

【0282】

次に、サブCPU81は、再生状態管理格納領域（図14参照）のエントリパーツ番号格納領域の値に基づいて、ランプデータの再生中であるか否かを判断する（S603）。前述したように、ランプデータの再生中である場合には、再生状態管理格納領域のエントリパーツ番号格納領域に再生中のパーツデータを識別するためのパーツ番号が格納されている。一方、後述するように、ランプデータの再生中でない場合には、再生状態管理格納領域のエントリパーツ番号格納領域に特定値（例えば、「0」）が格納されている。

【0283】

このため、本実施形態におけるサブCPU81は、再生状態管理格納領域のエントリパーツ番号格納領域の値が特定値でなければ、ランプデータの再生中であると判断し、再生状態管理格納領域のエントリパーツ番号格納領域の値が特定値であれば、ランプデータの再生中でないと判断する。

10

【0284】

S603において、サブCPU81は、ランプデータの再生中であると判断した場合には（YES）、S604の処理を実行し、ランプデータの再生中でないと判断した場合には（NO）、S602の処理を実行する。

【0285】

S604において、サブCPU81は、ランプデータにおける再生中のパーツデータの属性データを取得する。次に、サブCPU81は、再生中のパーツデータの再生位置における輝度パターンがエンドブロック（例えば、「-1」）であるか否かを判断する（S605）。

20

【0286】

S605において、サブCPU81は、再生中のパーツデータの再生位置における輝度パターンがエンドブロックであると判断した場合には（YES）、S608の処理を実行し、エンドブロックでないと判断した場合には（NO）、S606の処理を実行する。

【0287】

S606において、サブCPU81は、再生中のパーツデータの再生位置における輝度パターンから制御データ（図8～図10参照）を生成する。前述したように、本実施形態において、制御データは、6ビット（0～63の範囲）で輝度を表し、輝度パターンは、8ビット（0～255の範囲）で輝度を表す。

30

【0288】

このため、サブCPU81は、輝度パターンにおける輝度値をビットシフトにより6ビットの輝度値に変換（8ビットデータである輝度値をLSB方向に2ビットシフト）して、変換した輝度値に基づいてシリアルバス通信用の通信回路のデータフォーマットに応じた制御データを生成する。すなわち、サブCPU81は、6ビットに変換した輝度値を制御データの該当する輝度データLに登録する。

【0289】

なお、本実施形態において、輝度パターンの輝度値を8ビットで表しているが、これは輝度値が8ビット又は7ビットに対応するドライバICを使用した場合であっても、輝度値を変換する処理を変更するだけで、輝度パターンを転用可能とするためであり、ドライバICの種類が異なってもランプデータを流用することが可能となる。

40

【0290】

S606の処理を実行した後、S607の処理を実行する。S607において、サブCPU81は、ランプ制御データ生成処理（S606）で生成した制御データをランプドライバ（ソフトウェア）に登録する。

【0291】

このように、制御データをランプドライバに登録することによって、ランプドライバの処理を実行するサブCPU81は、制御データをドライバICに送信する。S607の処理を実行した後、サブCPU81は、S602の処理を実行する。

【0292】

50

S 6 0 8 において、サブCPU 8 1 は、再生状態管理格納領域（図 1 4 参照）のエントリパーツ番号格納領域とパーツ番号格納領域とを基に、再生中のパーツデータの次のパーツデータがあるか否かを判断する。S 6 0 8 において、サブCPU 8 1 は、次のパーツデータがあると判断した場合には（YES）、S 6 0 9 の処理を実行し、次のパーツデータがないと判断した場合（例えば、図 1 6 の場合、再生中パーツデータがパーツ番号：8、又はエンドコード格納領域（図 1 4 参照）のエンドコードを検出）には（NO）、S 6 1 0 の処理を実行する。

【0293】

S 6 0 9 において、サブCPU 8 1 は、再生状態管理格納領域のエントリパーツ番号格納領域を次のパーツ番号格納領域の値に更新し、再生位置を次のパーツデータに移動させる。S 6 1 0 において、サブCPU 8 1 は、S 6 0 4 で取得した属性データがショット+チェーンであるか否かを判断する。

10

【0294】

S 6 1 0 において、サブCPU 8 1 は、属性データがショット+チェーンであると判断した場合には（YES）、属性検索頭出処理を実行する（S 6 1 1）。属性検索頭出処理において、サブCPU 8 1 は、再生状態管理格納領域（図 1 4 ~ 図 1 8 参照）に記憶された内容に基づいて、チェーン/ループ設定領域に「TRUE」が格納されていれば、ランブデータが表す識別情報の先頭（例えば、図 1 6 の場合、パーツ番号：1）から属性データがショット+チェーンを表すパーツデータを検索（例えば、図 1 6 の場合、パーツ番号：2）し、検出したパーツデータの先頭に再生位置を移動（例えば、図 1 4 のエントリパーツ番号格納領域の値が「2」がセット）させる。S 6 1 1 の処理を実行した後、サブCPU 8 1 は、S 6 0 2 の処理を実行する。

20

【0295】

S 6 1 0 において、属性データがショット+チェーンでないと判断した場合には（NO）、サブCPU 8 1 は、S 6 0 4 で取得した属性データがループであるか否かを判断する（S 6 1 2）。S 6 1 2 において、属性データがループでないと判断した場合には（NO）、サブCPU 8 1 は、S 6 0 2 の処理を実行する。

【0296】

S 6 1 2 において、属性データがループであると判断した場合には（YES）、サブCPU 8 1 は、パーツ内頭出処理を実行する（S 6 1 3）。属性検索頭出処理において、サブCPU 8 1 は、再生中のパーツデータの先頭に再生位置を移動させる。S 6 1 3 の処理を実行した後、サブCPU 8 1 は、サブCPU 8 1 は、S 6 0 2 の処理を実行する。

30

【0297】

なお、パーツ内頭出処理とは、属性データがループであり、パーツデータが「0」~「9」まで、昇順に変化するデータであれば、パーツデータの先頭である「0」に頭出しされ、「0」~「9」までを繰り返す、リピート機能である。

【0298】

[ランブデータ読み込み処理]

次に、図 3 1 を参照して、ランブデータ読み込み処理について説明する。

【0299】

まず、サブCPU 8 1 は、パチスロ 1 のエラーが検出されたか否かを判断する（S 6 3 1）。パチスロ 1 のエラーには、ホッパー 3 3（図 5 参照）が空になったことを表すホッパーエンティエラー、メダル補助庫 3 7（図 5 参照）に収容されたメダルが規定量に達したことを表す投入メダル補助収納庫満杯エラーなどがある。

40

【0300】

メインCPU 5 1 は、これらのエラーを表すエラーコマンドを副制御回路 4 2 にする。サブCPU 8 1 は、このエラーコマンドに基づきパチスロ 1 のエラーが検出されたか否かを判断する。

【0301】

S 6 3 1 において、サブCPU 8 1 は、パチスロ 1 のエラーが検出されたと判断した場

50

合には (Y E S)、 S 6 3 2 の処理を実行し、パチスロ 1 のエラーが検出されなかったと判断した場合には (N O)、 S 6 3 4 の処理を実行する。

【 0 3 0 2 】

S 6 3 2 において、サブ C P U 8 1 は、ランプデータの再生中であれば、再生中のランプデータのリクエスト番号をサブ R A M 8 3 に保存する。 S 6 3 2 の処理を実行した後、サブ C P U 8 1 は、エラー用のランプデータを登録する (S 6 3 3)。 S 6 3 3 の処理を実行した後、サブ C P U 8 1 は、 S 6 3 8 の処理を実行する。

【 0 3 0 3 】

S 6 3 4 において、サブ C P U 8 1 は、パチスロ 1 のエラーが解除されたか否かを判断する (S 6 3 4)。 S 6 3 4 において、サブ C P U 8 1 は、パチスロ 1 のエラーが解除されたと判断した場合には (Y E S)、 S 6 3 5 の処理を実行し、パチスロ 1 のエラーが解除されていないと判断した場合には (N O)、 S 6 3 7 の処理を実行する。

10

【 0 3 0 4 】

S 6 3 5 において、サブ C P U 8 1 は、 S 6 3 2 で保存したリクエスト番号に対応する中断復帰用ランプデータのリクエスト番号を特定する。 S 6 3 5 の処理を実行した後、サブ C P U 8 1 は、 S 6 3 5 で特定したリクエスト番号のエラー用のランプデータを登録する (S 6 3 6)。 S 6 3 6 の処理を実行した後、サブ C P U 8 1 は、 S 6 3 8 の処理を実行する。

【 0 3 0 5 】

S 6 3 7 において、サブ C P U 8 1 は、演出登録タスクの S 5 1 6 (図 2 3 参照) で登録されたランプデータが更新されたか否かを判断する (S 6 3 7)。 S 6 3 7 において、サブ C P U 8 1 は、ランプデータが更新されたと判断した場合には (Y E S)、 S 6 3 8 の処理を実行し、ランプデータが更新されていないと判断した場合には (N O)、ランプデータ読み込み処理を終了する。

20

【 0 3 0 6 】

S 6 3 8 において、サブ C P U 8 1 は、 S 6 3 3、 S 6 3 6 又は演出登録タスクの S 5 1 6 で登録されたランプデータを取得する。 S 6 3 8 の処理を実行した後、サブ C P U 8 1 は、 S 6 3 8 で取得したランプデータに基づきサブ R A M 8 3 に再生状態管理格納領域 (図 1 4 参照) を生成する再生データ生成処理を実行する (S 6 3 9)。

【 0 3 0 7 】

本実施形態におけるサブ C P U 8 1 は、ランプデータの再生を終了させる場合には、演出登録タスクの S 5 1 6 でブランクのランプデータを登録する。ブランクのランプデータを登録した場合には、サブ C P U 8 1 は、エントリパーツ番号格納領域に特定値 (例えば、「 0 」) が設定され、パーツ番号格納領域を含まない再生状態管理格納領域を S 6 3 9 で生成する。 S 6 3 9 の処理を実行した後、サブ C P U 8 1 は、ランプデータ読み込み処理を終了する。

30

【 0 3 0 8 】

[ランプ関数 R A M 転送処理]

次に、図 3 2 を参照して、ランプ関数 R A M 転送処理について説明する。

【 0 3 0 9 】

まず、サブ C P U 8 1 は、ランプデータ読み込み処理における再生データ生成処理 (図 3 1 の S 6 3 9 参照) を実行するためのプログラムをサブ R O M 8 2 からサブ R A M 8 3 に転送する (S 6 5 1)。

40

【 0 3 1 0 】

次に、サブ C P U 8 1 は、ランプ制御タスクにおけるランプ制御データ生成処理 (図 3 0 の S 6 0 6) を実行するためのプログラムをサブ R O M 8 2 からサブ R A M 8 3 に転送する (S 6 5 2)。

【 0 3 1 1 】

次に、サブ C P U 8 1 は、ランプドライバをサブ R O M 8 2 からサブ R A M 8 3 に転送する (S 6 5 3)。 S 6 5 3 の処理を実行した後、サブ C P U 8 1 は、ランプ関数 R A M

50

転送処理を終了する。

【0312】

サブCPU81は、上位プログラムであるランプデータ読み込み処理のプログラムから再生データ生成処理のプログラムを実行し、上位プログラムであるランプ制御タスクのプログラムからランプ制御データ生成処理のプログラムを実行する。また、サブCPU81は、ランプドライバとしての処理をサブRAM83に転送したプログラムにしたがって実行する。

【0313】

このように、サブCPU81は、ランプ関数RAM転送処理において、相対的に処理負荷が高い所定のプログラムである再生データ生成処理及びランプ制御データ生成処理を実行するためのプログラム並びにランプドライバをサブROM82からサブRAM83に転送して実行することによって、処理速度を向上させている。

10

【0314】

[ランプ制御タスク、ランプデータ読み込み処理及びランプ関数RAM転送処理の具体例]

図33を参照して、図30～図32を参照してそれぞれ説明したランプ制御タスク、ランプデータ読み込み処理及びランプ関数RAM転送処理について具体的に説明する。なお、図33において、ランプ制御タスク及びランプ関数RAM転送処理は、それぞれC言語で表されている。

【0315】

「LampRamTrans()」は、図32に示したランプ関数RAM転送処理を実行するためのプログラムである。サブROM82のアドレス「_LampPlayMake」で表される位置には、図31に示したランプデータ読み込み処理のステップS639の処理を実行するためのプログラムが格納されている。

20

【0316】

ランプ関数RAM転送処理のステップS651では、メモリコピー関数(memcpy)によって、サブROM82のアドレス「_LampPlayMake」からサブRAM83のアドレス「__LampPlayMake」で表される位置に、プログラムサイズ(sizeof(_LampPlayMake))分のメモリコピーが行われる。

【0317】

このように、図31に示したランプデータ読み込み処理のステップS639の処理を実行するためのプログラムがサブRAM83に転送される。なお、本実施形態において、サブRAM83のアドレス「__LampPlayMake」で表される位置には、当該プログラムを格納するための領域がランプ関数RAM転送処理の実行前に確保されていることとする。

30

【0318】

サブROM82のアドレス「_LampMakeData」で表される位置には、図30に示したランプ制御タスクのステップS606の処理を実行するためのプログラムが格納されている。ランプ関数RAM転送処理のステップS652では、メモリコピー関数(memcpy)によって、サブROM82のアドレス「_LampMakeData」からサブRAM83のアドレス「__LampMakeData」で表される位置に、プログラムサイズ(sizeof(_LampMakeData))分のメモリコピーが行われる。

40

【0319】

このように、図30に示したランプ制御タスクのステップS606の処理を実行するためのプログラムがサブRAM83に転送される。なお、本実施形態において、サブRAM83のアドレス「__LampMakeData」で表される位置には、当該プログラムを格納するための領域がランプ関数RAM転送処理の実行前に確保されていることとする。

【0320】

サブROM82のアドレス「_SetLampDriver」で表される位置には、図30に示したランプ制御タスクのステップS607の処理を実行するためのプログラムが格納されている。ランプ関数RAM転送処理のステップS593では、メモリコピー関数(memcpy)によ

50

って、サブROM 82のアドレス「_SetLampDriver」からサブRAM 83のアドレス「__SetLampDriver」で表される位置に、プログラムサイズ(sizeof(_SetLampDriver))分のメモリコピーが行われる。

【0321】

このように、図30に示したランプ制御タスクのステップS607の処理を実行するためのプログラムがサブRAM 83に転送される。なお、本実施形態において、サブRAM 83のアドレス「__SetLampDriver」で表される位置には、当該プログラムを格納するための領域がランプ関数RAM転送処理の実行前に確保されていることとする。

【0322】

「LampTask()」は、図30に示したランプ制御タスクを実行するためのプログラムである。なお、図33に示した「LampTask()」は、ステップS605～S607の処理を表し、他の処理の記載は省略されている。

10

【0323】

「LampTask()」において、「LampData」は、サブROM 82の輝度データの格納先のアドレスを表すポインタ変数として用いられ、「LampBuf」は、サブRAM 83のランプ制御データの格納先のアドレスを表すポインタ変数として用いられる。「if(*LampData!=EndBlock)」の判定文は、ステップS605の処理を表す。

【0324】

「LampPlayMake()」は、図32に示したランプ関数RAM転送処理のステップS651でサブRAM 83のアドレス「__LampPlayMake」に転送されたプログラム「__LampPlayMake()」のアクセス関数である。

20

【0325】

「__LampPlayMake()」は、引数「dat」が表す輝度パターンの格納先のアドレスが指定されると、アドレスが指定された輝度パターンをサブRAM 83の再生状態管理格納領域(図14参照)に読み込み、再生状態管理格納領域のアドレスを引数「buf」に格納する。

【0326】

「LampMakeData()」は、図32に示したランプ関数RAM転送処理のステップS652でサブRAM 83のアドレス「__LampMakeData」に転送されたプログラム「__LampMakeData()」のアクセス関数である。

30

【0327】

「__LampMakeData()」は、引数「dat」が表す輝度パターンの格納先のアドレスが指定されると、アドレスが指定された輝度パターンに基づいてランプ制御データをサブRAM 83に生成し、生成したランプ制御データの格納先のアドレスを引数「buf」に格納する。

【0328】

したがって、「LampTask()」における「LampMakeData()」は、図30に示したランプ制御タスクのステップS606の処理を実行し、輝度パターンの格納先のアドレスを表すポインタ変数「LampData」が指定されることで、ポインタ変数「LampBuf」にランプ制御データの格納先であるサブRAM 83のアドレスを格納する。

40

【0329】

「SetLampDriver()」は、図32に示したランプ関数RAM転送処理のステップS653でサブRAM 83のアドレス「__SetLampDriver」に転送されたプログラム「__SetLampDriver()」のアクセス関数である。「__SetLampDriver()」は、「buf」が表すランプ制御データの格納先であるサブRAM 83のアドレスが指定されると、指定されたアドレスをランプドライバに登録する。

【0330】

したがって、「LampTask()」における「SetLampDriver()」は、図30に示したランプ制御タスクのステップS607の処理を実行し、ランプ制御データの格納先のアドレスを表すポインタ変数「LampBuf」が指定されることで、ランプ制御データの格納先であるサ

50

ブ R A M 8 3 のアドレスをランブドライバに登録する。

【 0 3 3 1 】

[サブ C P U に実行させるプログラムの製造方法]

次に、図 3 4 を参照して、副制御回路 4 2 のサブ C P U に実行させるプログラムの製造方法について説明する。

【 0 3 3 2 】

本実施形態において、サブ C P U 8 1 は、サブ R O M 8 2 に記憶されている制御プログラムを読み込みながら実行している。一方で、一般的に、R O M の読み込み速度より R A M の読み込み速度の方が、半導体の特性として速い。

【 0 3 3 3 】

このため、サブ C P U 8 1 は、一部の処理負荷の高いプログラム（一般的に、「関数」又は「サブルーチン」などと呼称されている。）をサブ R A M 8 3 に転送し、サブ R A M 8 3 に転送したプログラムを読み込みながら実行することによって、一部の処理負荷の高いプログラムの実行速度を向上させている。

【 0 3 3 4 】

ただし、サブ R O M 8 2 に記憶されているプログラムをサブ R A M 8 3 に転送しただけでは、転送したプログラムに記憶されているアドレスがサブ R O M 8 2 内のアドレスを指しているため正常に動作させることはできない。

【 0 3 3 5 】

プログラムをサブ R O M 8 2 に記憶させる前に、サブ R A M 8 3 に転送するプログラム内のアドレスをサブ R A M 8 3 内のアドレスに変更する必要がある。本実施形態において、サブ R A M 8 3 に転送するプログラムは、図 2 9 を参照して説明したサブ R O M 8 2 のアドレス「_SoundMakeData」及び「_SoundBuf」に格納されるプログラムと、図 3 3 を参照して説明したサブ R O M 8 2 のアドレス「_LampPlayMake」、「_LampMakeData」及び「_LampBuf」に格納されるプログラムとを含む。

【 0 3 3 6 】

また、サブ R A M 8 3 に転送するプログラムを使用する上位側プログラムにおいて、サブ R A M 8 3 に転送するプログラムの呼び出しアドレスをサブ R A M 8 3 内のアドレスに変更する必要がある。

【 0 3 3 7 】

本実施形態において、サブ R A M 8 3 に転送するプログラムを使用する上位側プログラムは、図 2 9 に示したアクセス関数である「SoundMakeData()」及び「SetSoundDriver()」と、図 3 3 に示したアクセス関数である「LampPlayMake()」、「LampMakeData()」及び「SetLampDriver()」とを含む。

【 0 3 3 8 】

したがって、本実施形態におけるプログラムの製造方法は、サブ R A M 8 3 に転送するプログラムと、サブ R A M 8 3 に転送するプログラムを使用する上位側プログラムと（以下、総称して「アドレス変換対象プログラム」ともいう）をアドレス変換対象プログラム以外のプログラムと異なる方法で製造する。

【 0 3 3 9 】

図 3 4 に示すように、本実施形態におけるプログラムの製造方法は、アドレス変換対象プログラムの対象 C 言語ソース群をコンパイルする工程 P 1 を実行した後に、工程 P 1 によって生成されたアセンブラソース群に対して上述したアドレス（具体低には、ラベル）の変更や命令（いわゆる、ニーモニック）の変更を行う工程 P 2 を有する。

【 0 3 4 0 】

変更内容として、例えば、ジャンプ先のラベルを変更し、変更前のラベルに絶対番地でジャンプする命令を変更後のラベルへの相対番地へジャンプする命令に書き換える。この場合、元の C 言語ソースから生成されたアセンブラソースのラベルはサブ R O M 8 2 を指しているため、サブ R A M 8 3 にジャンプできるように、ラベルを変更するとともに、相対位置（命令のある今のアドレスからプラス方向、又はマイナス方向への相対バイト数）

10

20

30

40

50

へジャンプする命令に書き換える。

【0341】

また、本実施形態におけるプログラムの製造方法は、アドレス変換対象プログラム以外のプログラムの非対象C言語ソース群をコンパイルしてオブジェクトファイル（中間ファイル）を生成し、工程P2によってアドレスが変更されたRAM実行対応済みアセンブラソース群をアセンブルしてオブジェクトファイルを生成し、非対象C言語ソース群のオブジェクトファイルとRAM実行対応済みアセンブラソース群のオブジェクトファイルとを互いにリンクさせて最終的なプログラムを生成する工程P3と、工程P3によって生成されたプログラムをサブROM82に書き込む工程P4とを有する。

【0342】

[各種効果]

以上に説明したように、本発明の実施形態に係るパチスロ1は、実行する演出に応じたパターンデータをドライバICとのシリアルバス通信用の通信回路のデータフォーマットに応じて変換し、変換したパターンデータに基づく制御データをドライバICに送信することにより第1ランプ群111、第7ランプ群117、第8ランプ群118及びドットマトリクス部119などの各種ランプ群を制御するため、ドライバICの仕様に依存しないパターンデータによって各種ランプ群を制御することができる。

【0343】

特に、本発明の実施形態に係るパチスロ1は、実行する演出に応じたパターンデータに含まれるLEDの輝度値をドライバICに応じた輝度値に変換し、変換した輝度値に基づく制御データをドライバICに送信することにより第1ランプ群111、第7ランプ群117、第8ランプ群118及びドットマトリクス部119などの各種ランプ群を制御するため、ドライバICの仕様に依存しないパターンデータによって各種ランプ群を制御することができる。

【0344】

また、本発明の実施形態に係るパチスロ1は、制御データにおいて、LEDの輝度を設定するためのデータに、エラー検出用データEを同期させないことにより、データパターンの影響を受けることなく、サブCPU81及びドライバIC間のシリアルバス通信の異常を制御データの受信側で検出させることができる。

【0345】

また、本発明の実施形態に係るパチスロ1は、ドライバICによって駆動される全てのLEDの輝度を設定する第1モードと、ドライバICに駆動させるLEDを特定するチャンネル番号及びチャンネル番号によって特定されるLEDの輝度を設定する第2モードと、のいずれのモードでも制御データを送信することができる。

【0346】

このため、マトリクス状に配置されたドットマトリクス部119を制御するときのように、LEDを個別に制御する場合には、特定のLEDの輝度を設定する第2モードの方が、第1モードよりも制御データのデータ長が短くなる。

【0347】

一方、第1ランプ群111、第7ランプ群117及び第8ランプ群118を制御するときのように、LEDを全体的に制御する場合には、LEDを特定するチャンネル番号を必要としない第1モードの方が、第2モードよりも制御データのデータ長が短くなる。

【0348】

また、全てのLEDを消灯させる場合には、各LEDの輝度値の指定やLEDを特定するチャンネル番号を必要としない第3モードの方が、第1モード及び第2モードよりもデータ長が短くなる。

【0349】

このように、本発明の実施形態に係るパチスロ1は、ドライバICが駆動するLEDの用途に応じて、データ長が短くなる制御データのモードを選択することにより、サブCPU81及びドライバIC間の伝送負荷を低減させることができ、LEDの応答性を向上さ

10

20

30

40

50

せることができる。

【0350】

また、本発明の実施形態に係るパチスロ1は、サブCPU81に動作用のクロックをカウントさせることによって日時データを管理させる。したがって、サブCPU81は、日時データを取得するたびに、RTC86とシリアル通信を行う必要がなくなる。このように、本発明の実施形態に係るパチスロ1は、サブCPU81にかかる負荷を低減させることができる。

【0351】

また、本発明の実施形態に係るパチスロ1は、閏年であるときの日数と閏年でないときの日数とが月ごとに対応付けられている月別日テーブルを参照して日時データを更新するため、日時データを誤りなく更新することができる。

10

【0352】

また、本発明の実施形態に係るパチスロ1は、サブRAM83の日時格納領域に記憶された日時データが表す日時に算出した経過時間が加算されるまで、サブRAM83の日時格納領域に記憶された日時データに規定時間(1秒カウント)を加算することによって、処理負荷が高い除算処理を行わずに加算処理だけでサブRAM83の日時格納領域に記憶された日時データを更新するため、サブCPU81にかかる負荷を低減させることができる。

【0353】

また、本発明の実施形態に係るパチスロ1は、相対的に処理負荷が高いプログラムであるサウンドデータ生成処理を実行するためのプログラム、サウンドドライバ、再生データ生成処理及びランプ制御データ生成処理を実行するためのプログラム並びにランプドライバをサブRAM83に転送して実行することによって、サブCPU81の処理速度を向上させることができる。

20

【0354】

また、本発明の実施形態に係るパチスロ1は、相対的に処理負荷が高いプログラム以外のプログラムをサブROM82に記憶された状態で実行することによって、サブRAM83に必要な記憶容量を抑制するため、コストを抑制することができる。

【0355】

また、本発明の実施形態に係るパチスロ1は、パーツデータを組み合わせることによって、第1ランプ群111、第7ランプ群117、第8ランプ群118及びドットマトリクス部119などの各種ランプ群に多様な演出を実行させるため、パーツデータ単位で各種ランプ群の発光パターンの確認及び変更を行わせることができる。このように、本発明の実施形態に係るパチスロ1は、各種ランプ群の発光パターンの確認及び変更にかかる負担を低減することができる。

30

【0356】

また、本発明の実施形態に係るパチスロ1は、属性データが連続再生に設定されているパーツデータによって、第1ランプ群111、第7ランプ群117、第8ランプ群118及びドットマトリクス部119などの各種ランプ群に演出を繰り返し実行させるため、各種ランプ群による演出を長時間にわたって実行させる場合であっても、各種ランプ群の発光パターンの確認及び変更にかかる負担を低減することができる。

40

【0357】

また、本発明の実施形態に係るパチスロ1は、エラー状態から復帰した場合、属性データがショット+チェインを表していないパーツデータの識別情報が除かれた新たな演出データが表す識別情報の順序にしたがってパーツデータに含まれるパターンデータで第1ランプ群111、第7ランプ群117、第8ランプ群118及びドットマトリクス部119などの各種ランプ群を制御するため、エラー復帰時に演出データを最初から再生することにより遊技状態にそぐわない演出を実行してしまうことを防止することができる。

【0358】

[変形例]

50

本実施形態において、サブCPU 81は、電源投入時にRTC 86から取得した日時データをサブRAM 83に記憶し、サブRAM 83に記憶した日時データを動作のクロックをカウントすることによって更新する例について説明した。

【0359】

これに加え、サブCPU 81は、所定の条件が成立するたびに、RTC 86から日時データを取得し、サブRAM 83に記憶した日時データを調節（アジャスト）するようにしてもよい。所定の条件としては、カウントした値が所定値を超えたこと、スタートコマンドのような特定のコマンドを受信したこと、エラー状態から復帰したこと、又は、リセットボタン（不図示）が押下されたことなどが挙げられる。

【0360】

また、本実施形態において、相対的に処理負荷が高いプログラムであるサウンドデータ生成処理を実行するためのプログラム、サウンドドライバ、再生データ生成処理及びランプ制御データ生成処理を実行するためのプログラム並びにランプドライバを起動時（本実施形態においては、サブCPU 81の電源投入時に起動されるサウンド制御タスク及びランプ制御タスクの初期化処理として、それぞれ実行されるサウンド関数RAM転送処理及びランプ関数RAM転送処理）にサブRAM 83に予め転送しておいて、実行する例について説明した。

【0361】

これに対し、本発明の実施形態に係るパチスロ1は、相対的に処理負荷が高い各プログラムを実行することを契機として各プログラムをサブRAM 83に転送して実行するようにしてもよい。

【0362】

また、本発明の実施形態に係るパチスロ1は、相対的に処理負荷が高い各プログラムを実行する場合、各プログラムがサブRAM 83に転送されていなければ、各プログラムをサブRAM 83に転送して実行するようにしてもよい。

【0363】

このように構成することにより、本発明の実施形態に係るパチスロ1は、相対的に処理負荷が高い各プログラムを実行する場合、各プログラムがサブRAM 83に転送されていれば、サブRAM 83に転送済みのプログラムを実行するため、各プログラムをサブRAM 83に転送するサブCPU 81の処理を省くことができる。

【0364】

また、本発明の実施形態に係るパチスロ1は、相対的に処理負荷が高い各プログラムを転送する領域がサブRAM 83に確保されていればサブRAM 83に転送し、各プログラムを転送する領域がサブRAM 83に確保されていなければサブRAM 83に各プログラムを転送する領域を確保し、確保した領域に各プログラムを転送して実行するようにしてもよい。

【0365】

このように構成することにより、本発明の実施形態に係るパチスロ1は、相対的に処理負荷が高い各プログラムを確実にサブRAM 83に転送して実行するため、サブCPU 81の処理速度を確実に向上させることができる。

【0366】

また、本発明の実施形態に係るパチスロ1は、相対的に処理負荷が高い各プログラムのうち、一部のプログラムを起動時にサブRAM 83に予め転送しておき、他のプログラムを実行時にサブRAM 83に転送するようにしてもよい。

【0367】

例えば、本発明の実施形態に係るパチスロ1は、サウンドデータ生成処理を実行するためのプログラム、サウンドドライバ、再生データ生成処理及びランプ制御データ生成処理を実行するためのプログラム並びにランプドライバのうち、サウンドドライバ及びランプドライバを起動時にサブRAM 83に予め転送し、サウンドデータ生成処理、再生データ生成処理及びランプ制御データ生成処理を実行するためのプログラムを実行時にサブRAM

10

20

30

40

50

M 8 3 に転送するようにしてもよい。

【 0 3 6 8 】

また、本実施形態において、サブCPU 8 1 は、第 1 発光体群の LED を駆動するドライバIC には、第 2 ポートから制御データを第 2 モードで送信し、第 2 発光体群の LED を駆動するドライバIC には、第 1 ポートから制御データを第 1 モードで送信する例について説明した。

【 0 3 6 9 】

これに対し、サブCPU 8 1 は、第 1 発光体群の LED を駆動するドライバIC には、第 2 ポートから制御データを第 1 モードで送信し、第 2 発光体群の LED を駆動するドライバIC には、第 1 ポートから制御データを第 2 モードで送信するようにしてもよい。

10

【 0 3 7 0 】

また、サブCPU 8 1 は、第 1 発光体群の LED を駆動するドライバIC には、第 2 ポートから制御データを第 1 モードで送信し、第 2 発光体群の LED を駆動するドライバIC には、第 1 ポートから制御データを第 1 モードで送信するようにしてもよい。

【 0 3 7 1 】

また、サブCPU 8 1 は、第 1 発光体群の LED を駆動するドライバIC には、第 2 ポートから制御データを第 2 モードで送信し、第 2 発光体群の LED を駆動するドライバIC には、第 1 ポートから制御データを第 2 モードで送信するようにしてもよい。

【 0 3 7 2 】

また、サブCPU 8 1 は、演出登録タスク（図 2 3 の S 5 1 6 参照）等でランプデータを登録したときに、登録したランプデータから生成する制御データをどのモードで送信するかを決定するようにしてもよい。

20

【 0 3 7 3 】

また、サブROM 8 2 に格納されたランプデータ又はパーツデータに予めモードを設定しておき、サブCPU 8 1 は、ランプデータ又はパーツデータに予め設定されたモードで制御データを送信するようにしてもよい。

【 0 3 7 4 】

[その他、本発明に係る遊技機の拡張性]

上記実施形態のパチスロ 1 では、遊技者のメダルの投入操作（すなわち、手持ちのメダルをメダル投入口 1 3 に対して投入する操作、又は、クレジットされたメダルを MAX ベットボタン 1 4 或いは 1 ベットボタン 1 5 を操作して投入する操作）により遊技が開始され、遊技が終了したときにメダルの払い出しがある場合には、ホッパー 3 3 を駆動してメダル払出口 1 8 からメダルが払い出され、又は、クレジットされる形態について説明したが、本発明はこれに限定されるものではない。

30

【 0 3 7 5 】

例えば、遊技者によって遊技に必要な遊技媒体が投入され、それに基づいて遊技が行われ、その遊技の結果に基づいて特典が付与される（例えば、メダルが払い出される）形態の全てに対して、本発明を適用することができる。すなわち、物理的な遊技者の動作によって遊技媒体が投入され（掛けられ）、遊技媒体が払い出される形態のみならず、主制御回路（主基板 3 1 ）自体が、遊技者が保有する遊技媒体を電磁的に管理し、メダルレスで遊技を可能にする形態であってもよい。なお、この場合、遊技者が保有する遊技媒体を電磁的に管理するのは、主制御回路（主基板 3 1 ）に装着され（接続され）且つ遊技媒体を管理する遊技媒体管理装置であってもよい。

40

【 0 3 7 6 】

この場合、遊技媒体管理装置は、ROM 及び RWM（あるいは、RAM）を有し、遊技機に設けられる装置であって、図示しない外部の遊技媒体取扱装置と所定のインターフェースを介して双方向通信可能に接続されるものであり、遊技媒体の貸出動作（すなわち、遊技者が遊技媒体の投入操作を行う上で、必要な遊技媒体を提供する動作）或いは遊技媒体の払い出しに係る役に入賞（当該役が成立）した場合における遊技媒体の払出動作（すなわち、遊技者に対して遊技媒体の払い出しを行う上で、必要な遊技媒体を獲得させる動

50

作)、又は、遊技の用に供する遊技媒体を電磁的に記録する動作を行い得るものとすればよい。また、遊技媒体管理装置は、実際の遊技媒体数の管理のみならず、例えば、その遊技媒体数の管理結果に基づいて、保有する遊技媒体数を表示する保有遊技媒体数表示装置(不図示)をパチスロ1の前面に設け、この保有遊技媒体数表示装置に表示される遊技媒体数を管理するものであってもよい。すなわち、遊技媒体管理装置は、遊技者が遊技の用に供することができる遊技媒体の総数を電磁的方法により記録し、表示することができるものとすればよい。

【0377】

また、この場合、遊技媒体管理装置は、遊技者が、記録された遊技媒体数を示す信号を、外部の遊技媒体取扱装置に対して自由に送信させることができる性能(機能)を有することが望ましい。また、遊技媒体管理装置は、遊技者が直接操作する場合以外の場合には、記録された遊技媒体数を減ずることができない性能を有することが望ましい。また、遊技媒体管理装置と外部の遊技媒体取扱装置との間に外部接続端子板(不図示)が設けられる場合には、遊技媒体管理装置は、その外部接続端子板を介してでなければ、遊技者が、記録された遊技媒体数を示す信号を送信できない性能を有することが望ましい。

10

【0378】

遊技機には、上記の他、遊技者が操作可能な貸出操作手段、返却(精算)操作手段、外部接続端子板が設けられ、遊技媒体取扱装置には、紙幣等の有価価値の投入口、記録媒体(例えばICカード)の挿入口、携帯端末から電子マネー等の入金を行うための非接触通信アンテナ等、その他貸出操作手段、返却操作手段等の各種操作手段、遊技媒体取扱装置側外部接続端子板が設けられるようにしてもよい(いずれも不図示)。

20

【0379】

その際の遊技の流れとしては、例えば、遊技者が遊技媒体取扱装置に対し、上記いずれかの方法で有価価値を入金し、上記いずれかの貸出操作手段の操作に基づいて所定数の有価価値を減算し、遊技媒体取扱装置から遊技媒体管理装置に対し、減算した有価価値に対応する遊技媒体を増加させる。そして、遊技者は遊技を行い、さらに遊技媒体が必要な場合には上記操作を繰り返し行う。その後、遊技の結果、所定数の遊技媒体を獲得し、遊技を終了する際には、上記いずれかの返却操作手段を操作することにより遊技媒体管理装置から遊技媒体取扱装置に対し、遊技媒体数を送信し、遊技媒体取扱装置はその遊技媒体数を記録した記録媒体を排出する。また、遊技媒体管理装置は遊技媒体数を送信したときに、自身が記憶する遊技媒体数をクリアする。遊技者は排出された記録媒体を景品交換するために景品カウンター等に持って行くか、又は、記録された遊技媒体に基づいて他の遊技台で遊技を行うために遊技台を移動する。

30

【0380】

なお、上記例では、遊技媒体管理装置から全遊技媒体数を遊技媒体取扱装置に対して送信したが、遊技機又は遊技媒体取扱装置側で遊技者が所望する遊技媒体数のみを送信し、遊技者が所持する遊技媒体を分割して処理することとしてもよい。また、上記例では、遊技媒体取扱装置が記録媒体を排出することとしたが、現金又は現金等価物を排出するようにしてもよいし、携帯端末等に記憶させるようにしてもよい。また、遊技媒体取扱装置は遊技場の会員記録媒体を挿入可能とし、遊技媒体を会員記録媒体に貯留して、後日、該貯留された遊技媒体を用いて再遊技可能とするようにしてもよい。

40

【0381】

また、遊技機又は遊技媒体取扱装置において、図示しない所定の操作手段を操作することにより遊技媒体取扱装置又は遊技媒体管理装置に対し、遊技媒体又は有価価値のデータ通信をロックするロック操作を実行可能としてもよい。その際には、ワンタイムパスワード等の遊技者にしか知り得ない情報を設定することや遊技機又は遊技媒体取扱装置に設けられた撮像手段により遊技者を記憶するようにしてもよい。

【0382】

なお、遊技媒体管理装置は、上述のように、メダルレスでのみ遊技を可能とするものであってもよいし、物理的な遊技者の動作によって遊技媒体が投入され(掛けられ)、遊技

50

媒体が払い出される形態、及び、メダルレスで遊技を可能とする形態の両方の形態で遊技を可能とするものであってもよい。後者の場合には、遊技媒体管理装置が、上述のセレクトタ35やホッパー33を直接的に制御する方式を採用することもできるし、これらが主制御回路（主基板31）によって制御され、その制御結果が送信されることに基づいて、遊技者が遊技の用に供することができる遊技媒体の総数を電磁的方法により記録し且つ表示する制御を行い得る方式を採用することもできる。

【0383】

また、上記例では、遊技媒体管理装置を、パチスロに適用する場合について説明しているが、例えば、遊技球を用いるスロットマシンや封入式遊技機においても同様に遊技媒体管理装置を設け、遊技者の遊技媒体が管理されるようにすることもできる。

10

【0384】

上述した遊技媒体管理装置を設けた場合には、遊技媒体が物理的に遊技に供される場合に比べて、遊技機内部のセレクトタ35やホッパー33などの装置を減らすことができ、遊技機の原価及び製造コストを削減できるのみならず、遊技者が直接遊技媒体に接触しないようにすることもでき、遊技環境が改善され、騒音も減らすことができるとともに、装置を減らしたことにより遊技機の消費電力を減らすことも可能になる。また、上述した遊技媒体管理装置を設けた場合には、遊技媒体や遊技媒体の投入口や払出口を介した不正行為を防止することができる。すなわち、上述した遊技媒体管理装置を設けた場合には、遊技機をとりまく種々の環境を改善可能な遊技機を提供することが可能になる。

20

【0385】

[発明の要旨]

< 要旨 1 >

LEDをドットマトリクス状に配置して表示ユニットを構成する遊技機が特開2005-323768号公報に提案されている。

【0386】

LEDをドットマトリクス状に配置したもののように、複数の発光体により構成された発光部の発光制御を行うためには、個々の発光体を直接に制御する方式と、制御データをLEDドライバICなどの発光駆動手段に送信することにより、発光駆動手段に個々の発光体を駆動させる方式とがある。

30

【0387】

発光駆動手段の仕様は、発光駆動手段の製造メーカ、又は、同じ製造メーカにおけるシリーズによって異なることがある。このため、発光駆動手段を介して個々の発光体を制御する場合には、発光駆動手段に応じて、発光部の発光パターンを表すパターンデータを作成し直す必要がある。

【0388】

本発明は、このような課題を解決するためになされたもので、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することを目的とする。

【0389】

本発明に係る遊技機は、
 複数の発光体（LED）により構成された発光部（第1ランプ群111、第7ランプ群117、第8ランプ群118、ドットマトリクス部119）と、
 前記発光部を駆動するための発光駆動手段（ドライバIC）と、
 前記発光部の発光パターンを制御するための制御部（サブCPU81）と、
 前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデータ記憶手段（サブROM82）と、を備え、
 前記制御部は、
 前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発光パターンに応じたパターンデータを決定し、
 決定した前記パターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じ

40

50

た輝度値に変換し、

変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信する

構成を有している。

【0390】

この構成により、本発明に係る遊技機は、実行する発光パターンに応じたパターンデータに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

【0391】

なお、本発明に係る遊技機において、

前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと同期しない周期（例えば、6ビット周期のデータに対して9ビット周期）でエラー検出用データを挿入するようにしてもよい。

【0392】

この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定するためのデータに、エラー検出用データを同期させないことにより、データパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができる。

【0393】

本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することができる。

【0394】

<要旨2>

要旨1と同様な課題を解決するため、本発明に係る遊技機は、

複数の発光体（LED）により構成された発光部（第1ランプ群111、第7ランプ群117、第8ランプ群118、ドットマトリクス部119）と、

前記発光部を駆動するための発光駆動手段（ドライバIC）と、

前記発光部の発光パターンを制御するための制御部（サブCPU81）と、

前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデータ記憶手段（サブROM82）と、を備え、

前記制御部は、

前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発光パターンに応じたパターンデータを決定し、

決定した前記パターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じた輝度値に変換し、

変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信し、

前記制御データは、

前記発光駆動手段によって駆動される全ての発光体の輝度を設定する第1モードと、

前記発光駆動手段に駆動させる発光体を特定するチャンネル番号及び前記チャンネル番号によって特定される発光体の輝度を設定する第2モードと、のいずれのモードでも送信可能である

構成を有している。

【0395】

この構成により、本発明に係る遊技機は、実行する発光パターンに応じたパターンデータに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

【0396】

10

20

30

40

50

また、本発明に係る遊技機は、発光駆動手段によって駆動される全ての発光体の輝度を設定する第1モードと、発光駆動手段に駆動させる発光体を特定するチャンネル番号及びチャンネル番号によって特定される発光体の輝度を設定する第2モードと、のいずれのモードでも制御データを送信することができる。

【0397】

このため、発光体を個別に制御する場合には、特定の発光体の輝度を設定する第2モードの方が、第1モードよりも制御データのデータ長が短くなり、発光体を全体的に制御する場合には、発光体を特定するチャンネル番号を必要としない第1モードの方が、第2モードよりも制御データのデータ長が短くなる。

【0398】

したがって、本発明に係る遊技機は、発光駆動手段が駆動する発光体の用途に応じて、データ長が短くなる制御データのモードを選択することにより、制御部と発光駆動手段との間の伝送負荷を低減させることができ、発光体の応答性を向上させることができる。

【0399】

なお、本発明に係る遊技機において、

前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと同期しない周期（例えば、6ビット周期のデータに対して9ビット周期）でエラー検出用データを挿入するようにしてもよい。

【0400】

この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定するためのデータに、エラー検出用データを同期させないことにより、データパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができる。

【0401】

本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することができる。

【0402】

<要旨3>

要旨1と同様な課題を解決するため、本発明に係る遊技機は、

複数の発光体（LED）により構成された発光部（第1ランプ群111、第7ランプ群117、第8ランプ群118、ドットマトリクス部119）と、

前記発光部を駆動するための発光駆動手段（ドライバIC）と、

前記発光部の発光パターンを制御するための制御部（サブCPU81）と、

前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデータ記憶手段（サブROM82）と、を備え、

前記制御部は、

前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発光パターンに応じたパターンデータを決定し、

決定した前記パターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じた輝度値に変換し、

変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信し、

前記制御データは、

前記発光駆動手段によって駆動される全ての発光体の輝度を設定する第1モードと、

前記発光駆動手段に駆動させる発光体を特定するチャンネル番号及び前記チャンネル番号によって特定される発光体の輝度を設定する第2モードと、のいずれのモードでも送信可能であり、

前記複数の発光体は、

マトリクス状に配置された第1発光体群と、

前記第1発光体群に含まれない第2発光体群と、を含み、

10

20

30

40

50

前記制御部は、

前記第1発光体群の発光体を駆動する発光駆動手段には、前記制御データを前記第2モードで送信し、

前記第2発光体群の発光体を駆動する発光駆動手段には、前記制御データを前記第1モードで送信する

構成を有している。

【0403】

この構成により、本発明に係る遊技機は、実行する発光パターンに応じたパターンデータに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

10

【0404】

また、本発明に係る遊技機は、発光駆動手段によって駆動される全ての発光体の輝度を設定する第1モードと、発光駆動手段に駆動させる発光体を特定するチャンネル番号及びチャンネル番号によって特定される発光体の輝度を設定する第2モードと、のいずれのモードでも制御データを送信することができる。

【0405】

このため、マトリクス状に配置された第1発光体群に対しては、発光体を個別に制御するため、特定の発光体の輝度を設定する第2モードの方が、第1モードよりも制御データのデータ長が短くなる。

20

【0406】

一方、第1発光体群に含まれない第2発光体群に対しては、発光体を全体的に制御するため、発光体を特定するチャンネル番号を必要としない第1モードの方が、第2モードよりも制御データのデータ長が短くなる。

【0407】

したがって、本発明に係る遊技機は、発光駆動手段が駆動する発光体の用途に応じて、データ長が短くなる制御データのモードを選択することにより、制御部と発光駆動手段との間の伝送負荷を低減させることができ、発光体の応答性を向上させることができる。

【0408】

なお、本発明に係る遊技機において、

30

前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと同期しない周期（例えば、6ビット周期のデータに対して9ビット周期）でエラー検出用データを挿入するようにしてもよい。

【0409】

この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定するためのデータに、エラー検出用データを同期させないことにより、データパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができる。

【0410】

本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することができる。

40

【0411】

<要旨4>

要旨1と同様な課題を解決するため、本発明に係る遊技機は、

複数の発光体（LED）により構成された発光部（第1ランプ群111、第7ランプ群117、第8ランプ群118、ドットマトリクス部119）と、

前記発光部を駆動するための発光駆動手段（ドライバIC）と、

前記発光部の発光パターンを制御するための制御部（サブCPU81）と、

前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデータ記憶手段（サブROM82）と、を備え、

50

前記制御部は、

前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発
光パターンに応じたパターンデータを決定し、

決定した前記パターンデータを前記発光駆動手段に応じて変換し、

変換した前記パターンデータに基づく制御データを前記発光駆動手段にシリアルバス通
信により送信する

構成を有している。

【0412】

この構成により、本発明に係る遊技機は、実行する前記発光パターンに応じたパターン
データを発光駆動手段に応じて変換し、変換したパターンデータに基づく制御データを発
光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しな
いパターンデータによって発光部を制御することができる。

10

【0413】

なお、本発明に係る遊技機において、

前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと
同期しない周期（例えば、6ビット周期のデータに対して9ビット周期）でエラー検出用
データを挿入するようにしてもよい。

【0414】

この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定す
るためのデータに、エラー検出用データを同期させないことにより、データパターンの影
響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることが
できる。

20

【0415】

本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制
御することができる遊技機を提供することができる。

【0416】

<要旨5>

要旨1と同様な課題を解決するため、本発明に係る遊技機は、

複数の発光体（LED）により構成された発光部（第1ランプ群111、第7ランプ群
117、第8ランプ群118、ドットマトリクス部119）と、

30

前記発光部を駆動するための発光駆動手段（ドライバIC）と、

前記発光部の発光パターンを制御するための制御部（サブCPU81）と、

前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデー
タ記憶手段（サブROM82）と、を備え、

前記制御部は、

前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発
光パターンに応じたパターンデータを決定し、

決定した前記パターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じ
た輝度値に変換し、

変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信
し、

40

前記制御データは、

前記発光駆動手段によって駆動される全ての発光体の輝度を設定する第1モードと、

前記発光駆動手段に駆動させる発光体を特定するチャンネル番号及び前記チャンネル番号に
よって特定される発光体の輝度を設定する第2モードと、

前記発光駆動手段によって駆動される全ての発光体を消灯させるコマンドを含む第3モ
ードと、のいずれのモードでも送信可能である

構成を有している。

【0417】

この構成により、本発明に係る遊技機は、実行する発光パターンに応じたパターンデー

50

タに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

【0418】

また、本発明に係る遊技機は、発光駆動手段によって駆動される全ての発光体の輝度を設定する第1モードと、発光駆動手段に駆動させる発光体を特定するチャンネル番号及びチャンネル番号によって特定される発光体の輝度を設定する第2モードと、のいずれのモードでも制御データを送信することができる。

【0419】

このため、発光体を個別に制御する場合には、特定の発光体の輝度を設定する第2モードの方が、第1モードよりも制御データのデータ長が短くなり、発光体を全体的に制御する場合には、発光体を特定するチャンネル番号を必要としない第1モードの方が、第2モードよりも制御データのデータ長が短くなる。

10

【0420】

また、全ての発光体を消灯させる場合には、各発光体の輝度値の指定や発光体を特定するチャンネル番号を必要としない第3モードの方が、第1モード及び第2モードよりもデータ長が短くなる。

【0421】

したがって、本発明に係る遊技機は、発光駆動手段が駆動する発光体の用途に応じて、データ長が短くなる制御データのモードを選択することにより、制御部と発光駆動手段との間の伝送負荷を低減させることができ、発光体の応答性を向上させることができる。

20

【0422】

なお、本発明に係る遊技機において、

前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと同期しない周期（例えば、6ビット周期のデータに対して9ビット周期）でエラー検出用データを挿入するようにしてもよい。

【0423】

この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定するためのデータに、エラー検出用データを同期させないことにより、データパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができる。

30

【0424】

本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することができる。

【0425】

<要旨6>

演出制御CPUとRTC（Real Time Clock）とがI2C（Inter-Integrated Circuit）によるシリアルバス通信で接続された遊技機が特開2017-51385号公報に提案されている。

【0426】

上述したような従来の遊技機は、RTCから日時データを取得するために、I2Cなどによるシリアル通信を行う必要があり、演出制御CPUなどの演算処理手段に負荷がかかることがある。

40

【0427】

本発明は、このような課題を解決するためになされたもので、演算処理手段にかかる負荷を低減させることができる遊技機を提供することを目的とする。

【0428】

本発明に係る遊技機は、

演算処理を行う演算処理手段（サブCPU81）と、

前記演算処理手段を動作させるためのクロックを発生するクロック発生手段（発振回路

50

84)と、

前記演算処理手段が使用するためのデータを記憶できる演算記憶手段(サブRAM83)と、

前記演算処理手段にシリアル通信で接続され日時を計時する計時手段(RTC86)と、を備え、

前記演算処理手段は、

前記計時手段から日時データを取得して前記演算記憶手段に記憶する日時取得手段と、

前記クロック発生手段によって発生されたクロックをカウントするカウント手段(クロックカウンタ)と、

前記カウント手段によってカウントされた値に基づいて経過時間を算出する経過時間算出手段と、

前記経過時間算出手段によって算出された経過時間が規定の経過時間に達した場合に、前記経過時間算出手段によって算出された経過時間に応じた時間分、前記演算記憶手段に記憶された前記日時データを更新する日時データ更新手段と、を有する

構成を有している。

【0429】

この構成により、本発明に係る遊技機は、演算処理手段に動作用のクロックをカウントさせることによって日時データを管理させる。したがって、演算処理手段は、日時データを取得するたびに、計時手段とシリアル通信を行う必要がなくなる。このように、本発明に係る遊技機は、演算処理手段にかかる負荷を低減させることができる。

【0430】

なお、本発明に係る遊技機において、前記日時データ更新手段は、前記経過時間から規定の経過時間を減算し、減算結果が規定の経過時間未満となるまで前記日時データを更新する処理を繰り返すようにしてもよい。

【0431】

本発明によれば、演算処理手段にかかる負荷を低減させることができる遊技機を提供することができる。

【0432】

<要旨7>

要旨6と同様な課題を解決するため、本発明に係る遊技機は、

演算処理を行う演算処理手段(サブCPU81)と、

前記演算処理手段を動作させるためのクロックを発生するクロック発生手段(発振回路84)と、

前記演算処理手段が使用するためのデータを記憶できる演算記憶手段(サブRAM83)と、

前記演算処理手段にシリアル通信で接続され日時を計時する計時手段(RTC86)と

、閏年であるときの日数と閏年でないときの日数とが月ごとに対応付けられている月別日テーブルを記憶する月別日テーブル記憶手段(サブROM82)と、を備え、

前記演算処理手段は、

前記計時手段から日時データを取得して前記演算記憶手段に記憶する日時取得手段と、

前記クロック発生手段によって発生されたクロックをカウントするカウント手段と、

前記カウント手段によってカウントされた値に基づいて経過時間を算出する経過時間算出手段と、

前記経過時間算出手段によって算出された経過時間が規定の経過時間に達した場合に、前記経過時間算出手段によって算出された経過時間に応じた時間分、前記演算記憶手段に記憶された前記日時データを更新する日時データ更新手段と、を有し、

前記日時データ更新手段は、前記月別日テーブルを参照して前記日時データを更新する構成を有している。

【0433】

10

20

30

40

50

この構成により、本発明に係る遊技機は、演算処理手段に動作用のクロックをカウントさせることによって日時データを管理させる。したがって、演算処理手段は、日時データを取得するたびに、計時手段とシリアル通信を行う必要がなくなる。このように、本発明に係る遊技機は、演算処理手段にかかる負荷を低減させることができる。

【0434】

また、本発明に係る遊技機は、閏年であるときの日数と閏年でないときの日数とが月ごとに対応付けられている月別日テーブルを参照して日時データを更新するため、日時データを誤りなく更新することができる。

【0435】

なお、本発明に係る遊技機において、前記日時データ更新手段は、前記経過時間から規定の経過時間を減算し、減算結果が規定の経過時間未満となるまで前記日時データを更新する処理を繰り返すようにしてもよい。

【0436】

本発明によれば、演算処理手段にかかる負荷を低減させることができる遊技機を提供することができる。

【0437】

<要旨8>

要旨6と同様な課題を解決するため、本発明に係る遊技機は、演算処理を行う演算処理手段(サブCPU81)と、前記演算処理手段を動作させるためのクロックを発生するクロック発生手段(発振回路84)と、

前記演算処理手段が使用するためのデータを記憶できる演算記憶手段(サブRAM83)と、

前記演算処理手段にシリアル通信で接続され日時を計時する計時手段(RTC86)と、

を備え、

前記演算処理手段は、

前記計時手段から日時データを取得して前記演算記憶手段に記憶する日時記憶手段と、

前記クロック発生手段によって発生されたクロックをカウントするカウント手段と、

前記カウント手段によってカウントされた値に基づいて経過時間を算出する経過時間算

出手段と、

前記経過時間算出手段によって算出された経過時間が規定の経過時間に達した場合に、前記経過時間算出手段によって算出された経過時間に応じた時間分、前記演算記憶手段に記憶された前記日時データを更新する日時データ更新手段と、を有し、

前記日時データ更新手段は、前記演算記憶手段に記憶された前記日時データが表す日時に前記経過時間算出手段によって算出された経過時間が加算されるまで、前記演算記憶手段に記憶された前記日時データに所定時間を加算する

構成を有している。

【0438】

この構成により、本発明に係る遊技機は、演算処理手段に動作用のクロックをカウントさせることによって日時データを管理させる。したがって、演算処理手段は、日時データを取得するたびに、計時手段とシリアル通信を行う必要がなくなる。このように、本発明に係る遊技機は、演算処理手段にかかる負荷を低減させることができる。

【0439】

また、本発明に係る遊技機は、演算記憶手段に記憶された日時データが表す日時に経過時間算出手段によって算出された経過時間が加算されるまで、演算記憶手段に記憶された日時データに所定時間を加算することによって、処理負荷が高い除算処理を行わずに加算処理だけで演算記憶手段に記憶された日時データを更新するため、演算処理手段にかかる負荷を低減させることができる。

【0440】

なお、本発明に係る遊技機において、前記日時データ更新手段は、前記経過時間から規

10

20

30

40

50

定の経過時間を減算し、減算結果が規定の経過時間未満となるまで前記日時データを更新する処理を繰り返すようにしてもよい。

【0441】

本発明によれば、演算処理手段にかかる負荷を低減させることができる遊技機を提供することができる。

【0442】

<要旨9>

ROMに格納されたプログラムをCPUに実行させる遊技機が特開2000-296223号公報に提案されている。また、ROMに格納されたプログラムを起動時のブート処理によりRAMに転送して、RAMに転送したプログラムをCPUに実行させる遊技機が特開2008-148891号公報に提案されている。

10

【0443】

ROMに格納されたプログラムをCPUなどの演算処理手段に実行させる従来の遊技機は、プログラムを実行するたびに演算処理手段がROMにアクセスするため、ROMのアクセス速度がボトルネックとなり、演算処理手段の処理速度を低下させていた。一方、RAMに転送したプログラムをCPUなどの演算処理手段に実行させる従来の遊技機は、必要なRAMの容量が増加するため、コストがかかってしまう。

【0444】

本発明は、このような課題を解決するためになされたもので、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することを目的とする。

20

【0445】

本発明に係る遊技機は、
演算処理を行う演算処理手段(サブCPU81)と、
前記演算処理手段を動作させるためのプログラム及びデータが記憶された第1記憶手段(サブROM82)と、
前記第1記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第2記憶手段(サブRAM83)と、を備え、
前記演算処理手段は、
前記第1記憶手段に記憶されたプログラムのうちの所定のプログラムを前記第2記憶手段に転送し、
前記所定のプログラムに対する上位プログラムから前記第2記憶手段に転送した前記所定のプログラムを実行する
構成を有している。

30

【0446】

この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラムを第2記憶手段に転送して実行するため、演算処理手段の処理速度を向上させることができる。

【0447】

また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第1記憶手段に記憶された状態で実行することによって、第2記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。

40

【0448】

なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第2記憶手段に転送する際には、第1の引数を前記第1記憶手段に記憶された所定のプログラムの先頭アドレスとし、第2の引数を前記第2記憶手段の転送先の先頭アドレスとし、第3の引数を前記第1記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。

【0449】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

50

【0450】

<要旨10>

要旨9と同様な課題を解決するため、本発明に係る遊技機は、演算処理を行う演算処理手段（サブCPU81）と、前記演算処理手段を動作させるためのプログラム及びデータが記憶された第1記憶手段（サブROM82）と、前記第1記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第2記憶手段（サブRAM83）と、を備え、前記演算処理手段は、前記第1記憶手段に記憶されたプログラムのうちの所定のプログラムを実行することを契機として前記所定のプログラムを前記第2記憶手段に転送し、前記所定のプログラムに対する上位プログラムから前記第2記憶手段に転送した前記所定のプログラムを実行する構成を有している。

10

【0451】

この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラムを実行することを契機として所定のプログラムを第2記憶手段に転送して実行することによって、演算処理手段の処理速度を向上させることができる。

【0452】

また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第1記憶手段に記憶された状態で実行することによって、第2記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。

20

【0453】

なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第2記憶手段に転送する際には、第1の引数を前記第1記憶手段に記憶された所定のプログラムの先頭アドレスとし、第2の引数を前記第2記憶手段の転送先の先頭アドレスとし、第3の引数を前記第1記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。

【0454】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

30

【0455】

<要旨11>

要旨9と同様な課題を解決するため、本発明に係る遊技機は、演算処理を行う演算処理手段（サブCPU81）と、前記演算処理手段を動作させるためのプログラム及びデータが記憶された第1記憶手段（サブROM82）と、前記第1記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第2記憶手段（サブRAM83）と、を備え、前記演算処理手段は、前記第1記憶手段に記憶されたプログラムのうちの所定のプログラムを前記第2記憶手段に予め転送しておき、前記所定のプログラムに対する上位プログラムから前記第2記憶手段に転送した前記所定のプログラムを実行する構成を有している。

40

【0456】

この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラムを第2記憶手段に予め転送しておいて実行することによって、演算処理手段の処理速度を向上させることができる。

【0457】

50

また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第1記憶手段に記憶された状態で実行することによって、第2記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。

【0458】

なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第2記憶手段に転送する際には、第1の引数を前記第1記憶手段に記憶された所定のプログラムの先頭アドレスとし、第2の引数を前記第2記憶手段の転送先の先頭アドレスとし、第3の引数を前記第1記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。

【0459】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

【0460】

<要旨12>

要旨9と同様な課題を解決するため、本発明に係る遊技機は、演算処理を行う演算処理手段（サブCPU81）と、前記演算処理手段を動作させるためのプログラム及びデータが記憶された第1記憶手段（サブROM82）と、前記第1記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第2記憶手段（サブRAM83）と、を備え、前記演算処理手段は、前記第1記憶手段に記憶されたプログラムのうちの所定のプログラムを実行する場合、前記所定のプログラムが前記第2記憶手段に転送されていなければ、前記所定のプログラムを前記第2記憶手段に転送し、前記所定のプログラムに対する上位プログラムから前記第2記憶手段に転送した前記所定のプログラムを実行する構成を有している。

【0461】

この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラム実行する場合、所定のプログラムが第2記憶手段に転送されていなければ、所定のプログラムを第2記憶手段に転送して実行することによって、演算処理手段の処理速度を向上させることができる。

【0462】

また、本発明に係る遊技機は、所定のプログラムを実行する場合、所定のプログラムが第2記憶手段に転送されていれば、第2記憶手段に転送済みの所定のプログラムを実行するため、所定のプログラムを第2記憶手段に転送する演算処理手段の処理を省くことができる。

【0463】

また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第1記憶手段に記憶された状態で実行することによって、第2記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。

【0464】

なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第2記憶手段に転送する際には、第1の引数を前記第1記憶手段に記憶された所定のプログラムの先頭アドレスとし、第2の引数を前記第2記憶手段の転送先の先頭アドレスとし、第3の引数を前記第1記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。

【0465】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

10

20

30

40

50

【 0 4 6 6 】

< 要旨 1 3 >

要旨 9 と同様な課題を解決するため、本発明に係る遊技機は、
演算処理を行う演算処理手段（サブ CPU 8 1）と、
前記演算処理手段を動作させるためのプログラム及びデータが記憶された第 1 記憶手段（サブ ROM 8 2）と、
前記第 1 記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第 2 記憶手段（サブ RAM 8 3）と、を備え、
前記演算処理手段は、
前記第 1 記憶手段に記憶されたプログラムのうちの所定のプログラムを転送する領域が
前記第 2 記憶手段に確保されていれば前記所定のプログラムを前記第 2 記憶手段に転送し

10

、
前記所定のプログラムを転送する領域が前記第 2 記憶手段に確保されていなければ前記第 2 記憶手段に前記所定のプログラムを転送する領域を確保し、確保した領域に前記所定のプログラムを転送し、

前記所定のプログラムに対する上位プログラムから前記第 2 記憶手段に転送した前記所定のプログラムを実行する

構成を有している。

【 0 4 6 7 】

この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラムを第 2 記憶手段に転送して実行することによって、演算処理手段の処理速度を向上させることができる。

20

【 0 4 6 8 】

また、本発明に係る遊技機は、所定のプログラムを転送する領域が第 2 記憶手段に確保されていなければ第 2 記憶手段に所定のプログラムを転送する領域を確保し、確保した領域に所定のプログラムを転送して実行するため、演算処理手段の処理速度を確実に向上させることができる。

【 0 4 6 9 】

また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第 1 記憶手段に記憶された状態で実行することによって、第 2 記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。

30

【 0 4 7 0 】

なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第 2 記憶手段に転送する際には、第 1 の引数を前記第 1 記憶手段に記憶された所定のプログラムの先頭アドレスとし、第 2 の引数を前記第 2 記憶手段の転送先の先頭アドレスとし、第 3 の引数を前記第 1 記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。

【 0 4 7 1 】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

40

【 0 4 7 2 】

< 要旨 1 4 >

LED をドットマトリクス状に配置して表示ユニットを構成する遊技機が特開 2 0 0 5 - 3 2 3 7 6 8 号公報に提案されている。

【 0 4 7 3 】

LED をドットマトリクス状に配置したもののように、複数の発光体により構成された発光部の発光制御を行うためには、全ての発光パターンを ROM 等に記憶しておき、発光パターンを適宜更新する。このため、演出内容に応じた大量の発光パターンを用意しておく必要があり、その確認及び変更が開発者並びに確認者の負担となる。

【 0 4 7 4 】

50

本発明は、このような課題を解決するためになされたもので、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供することを目的とする。

【0475】

本発明に係る遊技機は、

複数の発光体（LED）により構成された発光部（第1ランプ群111、第7ランプ群117、第8ランプ群118、ドットマトリクス部119）と、

前記発光部の発光パターンを制御するための制御部（サブCPU81）と、

前記発光部の前記発光パターンを制御するための複数の発光データ及び複数のパーツデータが記憶されたパターンデータ記憶手段（サブROM82）と、を備え、

前記パーツデータは、

前記パーツデータを識別するための識別情報と、

前記発光部の発光パターンを表すパターンデータと、

前記パーツデータの属性を示す属性データと、を含み、

前記発光データは、前記パーツデータの識別情報の順序を表し、

前記制御部は、

前記パターンデータ記憶手段に記憶された複数の発光データから、発光データを決定し、決定した発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、

前記発光データにおける最後の識別情報に対応するパーツデータに含まれるパターンデータで前記発光部を制御した後、前記発光データにおける最後の識別情報に対応するパーツデータに含まれる前記属性データが連続再生を表すことを条件として、前記発光データが表す識別情報の先頭から前記属性データが連続再生を表すパーツデータを検索し、検出したパーツデータから前記発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御する

構成を有している。

【0476】

この構成により、本発明に係る遊技機は、パーツデータを組み合わせることによって、発光部に多様な演出を実行させるため、パーツデータ単位で発光部の発光パターンの確認及び変更を行わせることができる。このように、本発明に係る遊技機は、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【0477】

また、本発明に係る遊技機は、属性データが連続再生に設定されているパーツデータによって発光部に演出を繰り返し実行させるため、発光部による演出を長時間にわたって実行させる場合であっても、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【0478】

本発明によれば、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供することができる。

【0479】

<要旨15>

要旨14と同様な課題を解決するため、本発明に係る遊技機は、

複数の発光体（LED）により構成された発光部（第1ランプ群111、第7ランプ群117、第8ランプ群118、ドットマトリクス部119）と、

前記発光部の発光パターンを制御するための制御部（サブCPU81）と

前記発光部の発光パターンを制御するための複数の発光データ及び複数のパーツデータが記憶された発光データ記憶手段（サブROM82）と、を備えた遊技機であって、

前記パーツデータは、

前記パーツデータを識別するための識別情報と、

前記発光部の発光パターンを表すパターンデータと、

前記パーツデータの属性を示す属性データと、を含み、

10

20

30

40

50

前記発光データは、前記パーツデータの識別情報の順序を表し、
前記制御部は、

前記発光データ記憶手段に記憶された複数の発光データから、発光データを決定し、決定した発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、

前記発光データにおける最後の識別情報に対応するパーツデータに含まれるパターンデータで前記発光部を制御した後、前記発光データにおける最後の識別情報に対応するパーツデータに含まれる前記属性データが連続再生を表すことを条件として、前記発光データが表す識別情報の先頭から前記属性データが連続再生を表すパーツデータを検索し、検出したパーツデータから前記発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、

前記遊技機がエラー状態から復帰した場合、前記遊技機が前記エラー状態となったときに実行していたエラー状態データが表すパーツデータの識別情報の順序から前記属性データが前記連続再生を表していないパーツデータの識別情報が除かれた発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御する

構成を有している。

【0480】

この構成により、本発明に係る遊技機は、パーツデータを組み合わせることによって、発光部に多様な演出を実行させるため、パーツデータ単位で発光部の発光パターンの確認及び変更を行わせることができる。このように、本発明に係る遊技機は、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【0481】

また、本発明に係る遊技機は、属性データが連続再生に設定されているパーツデータによって発光部に演出を繰り返し実行させるため、発光部による演出を長時間にわたって実行させる場合であっても、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【0482】

また、本発明に係る遊技機は、エラー状態から復帰した場合、属性データが連続再生を表していないパーツデータの識別情報が除かれた新たな演出データが表す識別情報の順序にしたがってパーツデータに含まれるパターンデータで発光部を制御するため、エラー復帰時に演出データを最初から再生することにより遊技状態にそぐわない演出を実行してしまうことを防止することができる。

【0483】

本発明によれば、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供することができる。

【0484】

<要旨16>

要旨14と同様な課題を解決するため、本発明に係る遊技機は、

複数の発光体（LED）により構成された発光部（第1ランプ群111、第7ランプ群117、第8ランプ群118、ドットマトリクス部119）と、

前記発光部を駆動するための発光駆動手段（LEDドライバIC）と、

前記発光部の発光パターンを制御するための制御部（サブCPU81）と、

前記発光部の前記発光パターンを制御するための複数の発光データ及び複数のパーツデータが記憶された発光データ記憶手段（サブROM82）と、を備え、

前記パーツデータは、

前記パーツデータを識別するための識別情報と、

前記発光部の発光パターンを表すパターンデータと、

前記パーツデータの属性を示す属性データと、を含み、

前記発光データは、前記パーツデータの識別情報の順序を表し、

10

20

30

40

50

前記制御部は、

前記発光データ記憶手段に記憶された複数の発光データから、発光データを決定し、決定した発光データが表す識別情報の順序にしたがって前記パーツデータのパターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信することによって前記発光部を制御し、

前記発光データにおける最後の識別情報に対応するパーツデータに含まれるパターンデータで前記発光部を制御した後、前記発光データにおける最後の識別情報に対応するパーツデータに含まれる前記属性データが連続再生を表すことを条件として、前記発光データが表す識別情報の先頭から前記属性データが連続再生を表すパーツデータを検索し、検出したパーツデータから前記発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御する

構成を有している。

【0485】

この構成により、本発明に係る遊技機は、パーツデータを組み合わせることによって、発光部に多様な演出を実行させるため、パーツデータ単位で発光部の発光パターンの確認及び変更を行わせることができる。このように、本発明に係る遊技機は、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【0486】

また、本発明に係る遊技機は、属性データが連続再生に設定されているパーツデータによって発光部に演出を繰り返し実行させるため、発光部による演出を長時間にわたって実行させる場合であっても、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【0487】

また、本発明に係る遊技機は、パターンデータに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

【0488】

本発明によれば、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供することができる。

【符号の説明】

【0489】

1 パチスロ（遊技機）

81 サブCPU（制御部、演算処理手段、日時取得手段、カウント手段、経過時間算出手段、日時データ更新手段）

82 サブROM（パターンデータ記憶手段、月別日テーブル記憶手段、第1記憶手段、演出データ記憶手段）

83 サブRAM83（演算記憶手段、第2記憶手段）

84 発振回路（クロック発生手段）

86 RTC（計時手段）

111 第1ランプ群（発光部、第2発光体群）

117 第7ランプ群（発光部、第2発光体群）

118 第8ランプ群（発光部、第2発光体群）

119 ドットマトリクス部（発光部、第1発光体群）

121a～121b、127a～127d、128a～128c、129a～129g
ドライバIC（発光駆動手段）

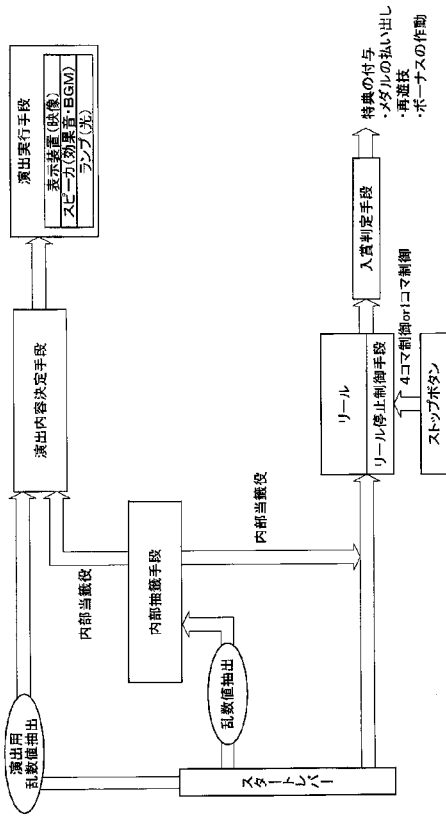
10

20

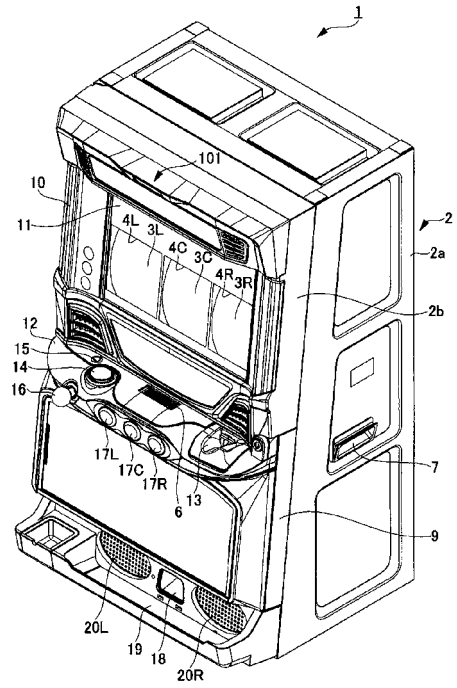
30

40

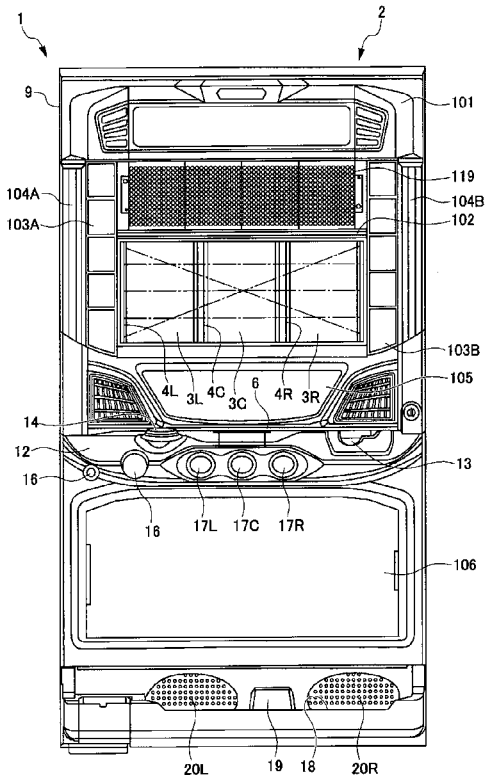
【 図 1 】



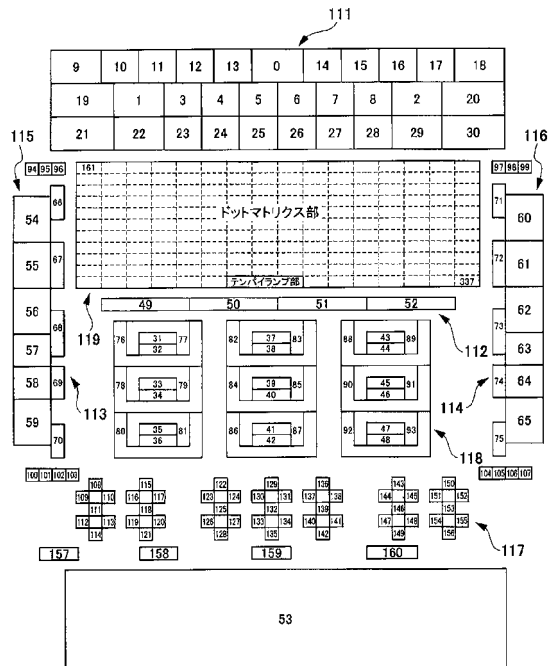
【 図 2 】



【 図 3 】



【 図 4 】



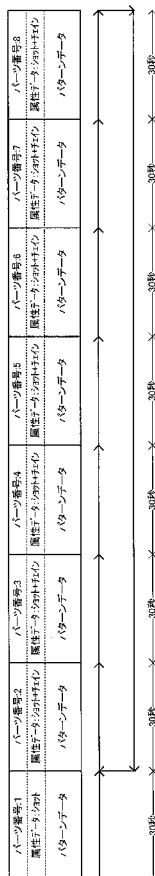
【 図 1 4 】

再生状態管理格納領域	
格納領域	内容
エントリパーツ番号格納領域	再生中のパーツ番号
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
エンドコード格納領域	エンドコード
チェイン/ループ設定領域	チェイン又はループの有無

【 図 1 5 】

格納領域	値
エントリパーツ番号格納領域	1
パーツ番号格納領域	1
パーツ番号格納領域	2
パーツ番号格納領域	3
パーツ番号格納領域	4
パーツ番号格納領域	5
パーツ番号格納領域	6
パーツ番号格納領域	7
パーツ番号格納領域	8
エンドコード格納領域	エンドコード
チェイン/ループ設定領域	TRUE

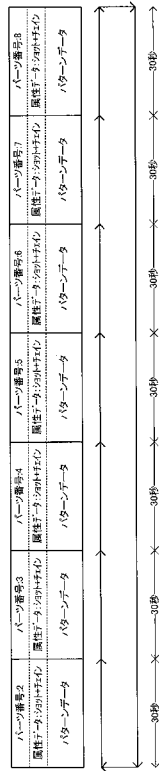
【 図 1 6 】



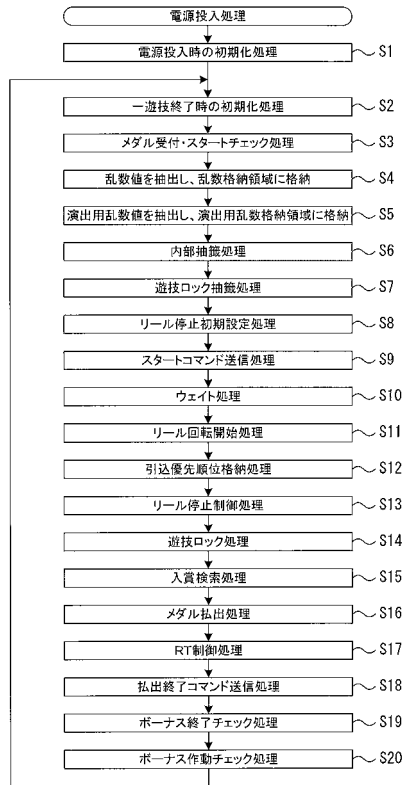
【 図 1 7 】

格納領域	値
エントリパーツ番号格納領域	2
パーツ番号格納領域	2
パーツ番号格納領域	3
パーツ番号格納領域	4
パーツ番号格納領域	5
パーツ番号格納領域	6
パーツ番号格納領域	7
パーツ番号格納領域	8
エンドコード格納領域	エンドコード
チェイン/ループ設定領域	TRUE

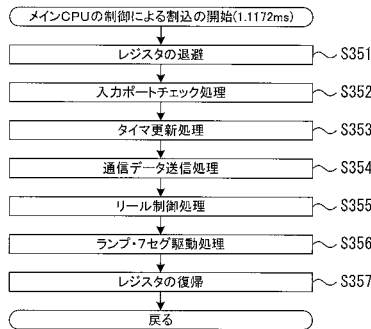
【 図 1 8 】



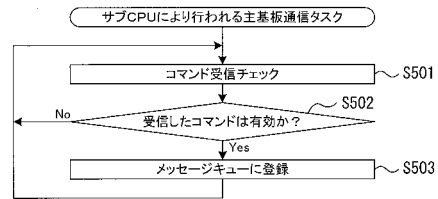
【 図 1 9 】



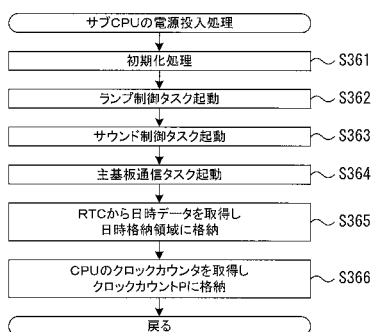
【 図 2 0 】



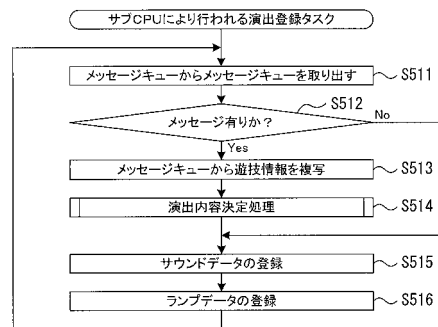
【 図 2 2 】



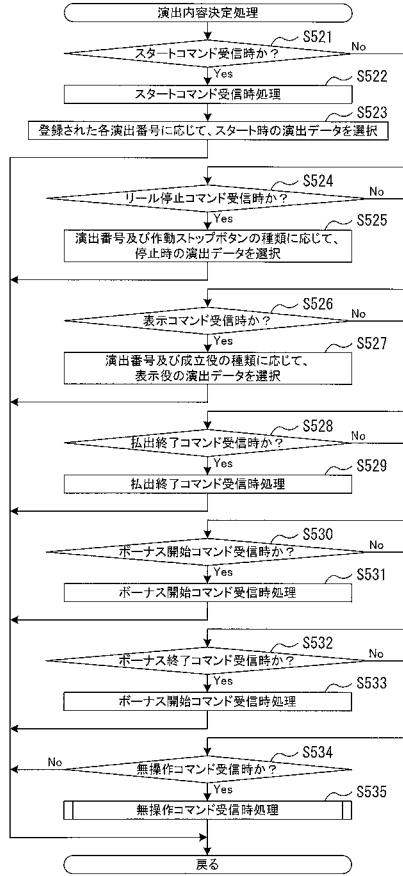
【 図 2 1 】



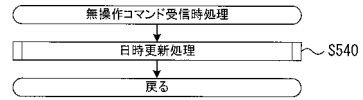
【 図 2 3 】



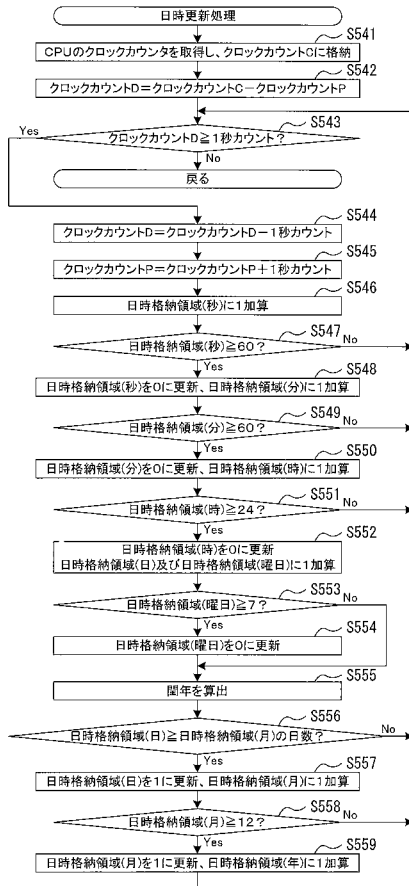
【 図 2 4 】



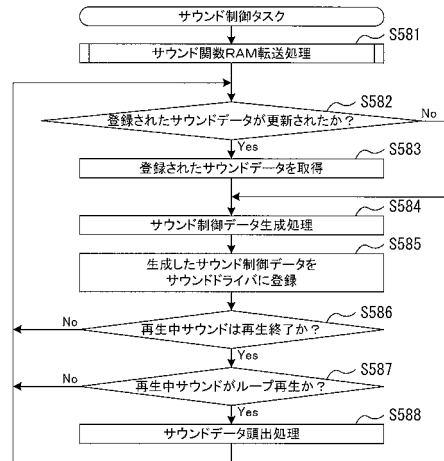
【 図 2 5 】



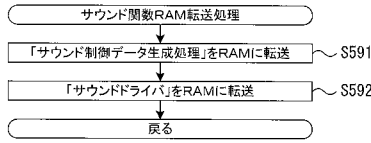
【 図 2 6 】



【 図 2 7 】



【 図 2 8 】



【 図 2 9 】

```

//サウンド関数RAM転送処理
void SoundRamTrans()
{
  // _SoundMakeDataをRAMの _SoundMakeData領域にコピー
  memcpy(_SoundMakeData, _SoundMakeData, sizeof(_SoundMakeData)); //S591
  // _SetSoundDriverをRAMの _SetSoundDriver領域にコピー
  memcpy(_SetSoundDriver, _SetSoundDriver, sizeof(_SetSoundDriver)); //S592
}

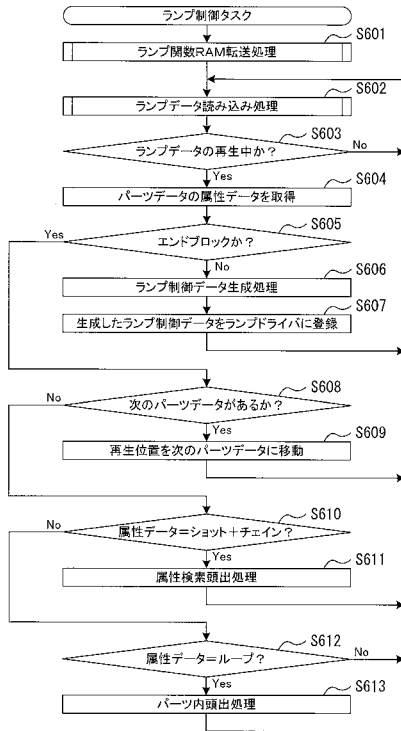
//サウンド制御タスク
void SoundTask()
{
  unsigned int index=0; //サウンドデータのリスト番号
  unsigned char *SoundData; //サウンドデータの保存領域
  unsigned char *SoundBuf; //サウンド制御データの保存領域

  省略
  GetSoundData(index, SoundData); //登録されたサウンドデータを取得(S583)
  SoundMakeData(SoundData, SoundBuf); //サウンド制御データ生成処理(S584)
  SetSoundDriver(SoundBuf); //サウンド制御データをサウンドドライバに登録(S585)
  省略
}

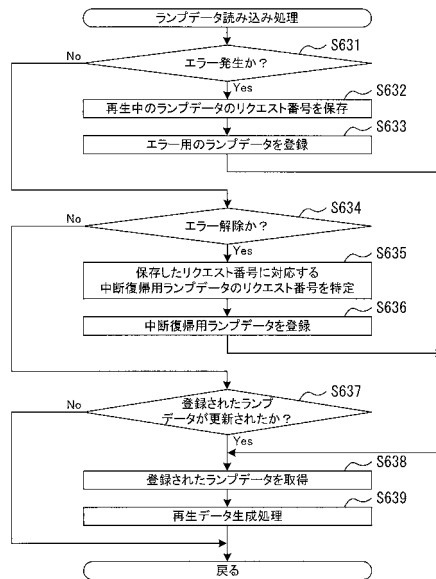
//RAMへコピーされたサウンド制御データ生成処理へのアクセス関数
void SoundMakeData(unsigned char *dat, unsigned char *buf)
{
  _SoundMakeData(dat, buf); //RAMへコピーされたサウンド制御データ生成処理
}

//RAMへコピーされたサウンドドライバへのアクセス関数
void SetSoundDriver(unsigned char *buf)
{
  _SetSoundDriver(buf); //サウンドドライバへサウンド制御データを登録
}
  
```

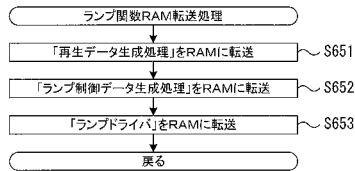
【 図 3 0 】



【 図 3 1 】



【 図 3 2 】



【 図 3 3 】

```

//ランプ関数RAM転送処理
void LampRamTrans(){
  //「_LampPlayMake」をRAMの「_LampPlayMake」領域にコピー
  memcpy(_LampPlayMake,_LampPlayMake,sizeof(_LampPlayMake)); //S651
  //「_LampMakeData」をRAMの「_LampMakeData」領域にコピー
  memcpy(_LampMakeData,_LampMakeData,sizeof(_LampMakeData)); //S652
  //「_SetLampDriver」をRAMの「_SetLampDriver」領域にコピー
  memcpy(_SetLampDriver,_SetLampDriver,sizeof(_SetLampDriver)); //S653
}

//ランプ制御タスク
void LampTask(){
  unsigned char *LampData; //輝度パターンの保存領域
  unsigned char *LampBuf; //ランプ制御データの保存領域

  省略
  if(*LampData!=EndBlock){ //S605
    LampMakeData(LampData, LampBuf); //ランプ制御データ生成処理(S606)
    SetLampDriver(LampBuf); //ランプ制御データをランプドライバに登録(S607)
  }
  省略
}

//RAMへコピーされた再生データ生成処理へのアクセス関数
void LampPlayMake(unsigned char *dat,unsigned char *buf){
  _LampPlayMake(dat, buf); //RAMへコピーされた再生データ生成処理
}

//RAMへコピーされたランプ制御データ生成処理へのアクセス関数
void LampMakeData(unsigned char *dat,unsigned char *buf){
  _LampMakeData(dat,buf); //RAMへコピーされたランプ制御データ生成処理
}

//RAMへコピーされたランプドライバへのアクセス関数
void SetLampDriver(unsigned char *buf){
  _SetLampDriver(buf); //ランプドライバへランプ制御データを登録
}
  
```

【 図 3 4 】

