

(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2007-221228

(P2007-221228A)

(43) 公開日 平成19年8月30日(2007.8.30)

(51) Int. Cl.		F I		テーマコード (参考)
HO4M	1/73	(2006.01)	HO4M 1/73	5K027
HO4B	7/26	(2006.01)	HO4B 7/26	5K067

審査請求 未請求 請求項の数 6 O L (全 9 頁)

(21) 出願番号 特願2006-36645 (P2006-36645)
 (22) 出願日 平成18年2月14日 (2006.2.14)

(71) 出願人 000004237
 日本電気株式会社
 東京都港区芝五丁目7番1号
 (74) 代理人 100105511
 弁理士 鈴木 康夫
 (74) 代理人 100109771
 弁理士 白田 保伸
 (72) 発明者 柚ノ木 和幸
 東京都港区芝五丁目7番1号 日本電気株式会社社内
 Fターム(参考) 5K027 AA11 BB17 FF22
 5K067 AA43 BB04 CC21 EE02 FF02
 FF23 FF31 KK17

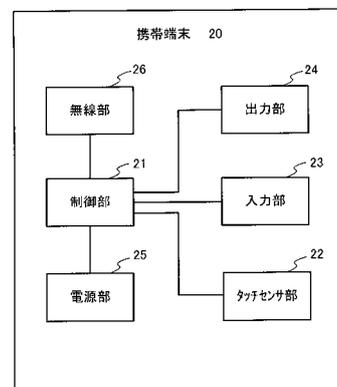
(54) 【発明の名称】 携帯端末装置

(57) 【要約】

【課題】 仮想マシン機能により実行されるアプリケーションソフト終了処理忘れによるバッテリーの消耗を抑制する。

【解決手段】 携帯電話機等の携帯端末20に、該端末にユーザーが触れていることを検知することができるタッチセンサ部22を設け、制御部21にタッチセンサ部22から接触信号が出力されているときは携帯端末20がユーザーにより使用中であると判定し、非接触信号が出力されているときには携帯端末20は現在使用されていないと判定する。この携帯端末20上でアプリケーションソフトが実行されているときに、タッチセンサ部22から非接触信号が出力されたとき、制御部21は実行中のアプリケーションソフトの実行を一時中断することにより、携帯端末の消費電力を抑える。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

携帯電話機機能に加えて、仮想マシン機能と該仮想マシン機能により実行されるアプリケーションプログラムが実装された携帯端末装置において、

前記携帯端末装置の使用状態と不使用状態を検出する使用 / 不使用検出手段と、

前記アプリケーションプログラムが起動されている状態において前記使用状態検出手段により前記携帯端末装置の不使用状態が検出されたとき、前記アプリケーションプログラムの実行を中断させる制御を行うアプリケーションプログラム実行制御手段と、
を備えていることを特徴とする携帯端末装置。

【請求項 2】

前記アプリケーションプログラム実行制御手段は、前記アプリケーションプログラムの実行が中断している状態において前記使用状態検出手段により前記携帯端末装置の使用状態が検出されたとき、前記中断状態のアプリケーションプログラムの実行を再開させる制御を行う機能を有していることを特徴とする請求項 1 に記載の携帯端末装置。

【請求項 3】

前記アプリケーションプログラムが起動されている状態において前記使用状態検出手段により前記携帯端末装置の不使用状態が検出されている間、前記携帯端末装置の表示部にデフォルト画面を表示する処理を実行し、前記アプリケーションプログラムの実行が中断している状態において前記使用状態検出手段により前記携帯端末装置の使用状態が検出されたとき、前記デフォルト画面の表示を停止する処理を実行する手段を有していることを特徴とする請求項 2 に記載の携帯端末装置。

【請求項 4】

前記携帯端末装置のキーボードが配置されている筐体の側面に配置されて、ユーザーの接触状態を検知するタッチセンサを備えており、前記使用 / 不使用検出手段は、前記タッチセンサにより前記ユーザーの接触状態が検知されているときには使用状態と判定し、前記ユーザーの接触状態が検知されていないときには不使用状態と判定する機能を有していることを特徴とする請求項 1 ~ 3 のいずれか 1 項に記載の携帯端末装置。

【請求項 5】

前記携帯端末装置は、ヒンジ部により折り畳み可能に構成された 2 つの筐体を有しており、前記使用 / 不使用検出手段は、前記 2 つの筐体がヒンジ部により開かれた状態と折り畳まれた状態を検知する手段を備え、開かれた状態のとき使用状態と判定し、折り畳まれた状態のとき不使用状態と判定する機能を有していることを特徴とする請求項 1 ~ 3 のいずれか 1 項に記載の携帯端末装置。

【請求項 6】

携帯電話機機能と、仮想マシン機能により実行されるアプリケーションプログラムが実装された携帯端末装置が備える制御部に用いるコンピュータで実行される制御用プログラムであって、

ユーザーの前記携帯端末装置への接触状態を判定する第 1 の処理と、前記アプリケーションプログラム実行中に、前記第 1 の処理 1 によりユーザーの接触状態が解除されたときと判定されたときに前記実行中のアプリケーションプログラムの実行を一時中断する第 2 の処理と、該第 2 の処理により前記アプリケーションプログラム実行が中断されている状態において、前記第 1 の処理によりユーザー前記携帯端末装置への接触状態が検知されたときと判定されたときに中断中の前記アプリケーションプログラムの実行を再開する第 3 の処理を前記コンピュータに実行させるためのプログラム。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、仮想マシン機能を実装することにより、IPサーバ等からダウンロードした種々のアプリケーションプログラムを、端末上で実行する機能を有する携帯電話機等の携

10

20

30

40

50

帯端末装置に関する。

【背景技術】

【0002】

最近の携帯電話機等の携帯端末装置は、電話機能あるいはメール機能等の携帯電話機として通常有している機能に加えて、Java(サンマイクロシステムズ社の登録商標)やBREW(Binary Runtime Environment for Wireless)仮想マシン機能を実装し、JavaやBREWで記述された、携帯電話機能とは直接関係しない様々なアプリケーションプログラムを格納し、端末上で実行できるようにしたものが多くなっている(特許文献1)。このアプリケーションプログラムを実行することにより、携帯端末装置のディスプレイ上に3次元動画からなる待ち受け画面を表示させたり、ゲームを行ったり、あるいは音楽を再生することができるようになっている。

10

【0003】

【特許文献1】特開2003-101628号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

このようなアプリケーションプログラムを実行可能な携帯電話機により、JavaやBREWアプリケーションを実行している途中で、たとえば、電車などから降りる際に携帯端末をそのまま閉じてバッグやポケットに入れてしまうなどして、JavaやBREWアプリケーションを終了させずにそのままの状態に放置した場合、JavaやBREWアプリケーションが起動し続けて端末のバッテリーを消費してしまうことがあり、いざ音声やメールの発信を行おうというときに携帯端末を使えない状態にしてしまうという問題があった。

20

【0005】

携帯電話や携帯端末は1回の充電で使用可能な時間が限られているので、バッテリー維持時間の向上を考えた場合、消費電力の大きいJavaやBREWで記述されたアプリケーションプログラムの実行時間を少しでも短くすることが必要である。

【0006】

本発明の目的は、上記問題点に鑑み、ユーザーがJavaやBREWで記述されたアプリケーションソフト終了処理を忘れた場合でもそのソフトを何らかの形で制御することにより、ソフト駆動時間を短縮し、バッテリー維持時間の向上を図ることが可能な手段を提供することにある。

30

【課題を解決するための手段】

【0007】

本発明は、携帯電話機機能に加えて、仮想マシン機能と該仮想マシン機能により実行されるアプリケーションプログラムが実装された携帯端末装置において、該携帯端末装置の使用状態と不使用状態を検出する使用/不使用検出手段と、前記アプリケーションプログラムが起動されている状態において前記使用状態検出手段により前記携帯端末装置の不使用状態が検出されたとき、前記アプリケーションプログラムの実行を中断させる制御を行うアプリケーションプログラム実行制御手段と、を備えていることを特徴とする。

40

【0008】

携帯端末装置の使用状態と不使用状態を検出する手段として種々の手段を採用することができるが、例えば、携帯端末のキーボードが配置されている筐体の側面にタッチセンサを配置し、ユーザーが手で携帯端末筐体を握っているときにユーザーの接触状態を検知して使用状態であると判定し、手から離れた非接触状態を検知して不使用状態であると判定する構成を採用することができる。そして、携帯端末上で実行されているJavaやBREWアプリケーションをタッチセンサでユーザーの接触を検知している状態のときのみ起動させ、非接触を検知している時にはJavaアプリケーションの実行を一時中断することにより、携帯端末の消費電流を抑えるようにする。

【0009】

50

あるいは、携帯端末装置が、ヒンジ部により折り畳み可能に構成された2つの筐体によって構成されている場合には、2つの筐体がヒンジ部により開かれた状態と折り畳まれた状態を検知する手段を備え、開かれた状態のとき使用状態と判定し、折り畳まれた状態のとき不使用状態と判定する構成を採用することもできる。

【0010】

また、前記アプリケーションプログラム実行制御手段は、前記アプリケーションプログラムの実行が中断している状態において前記使用/不使用検出手段により前記携帯端末装置の使用状態が検出されたとき、前記中断状態のアプリケーションプログラムの実行を自動的に再開させる制御を行う機能を備えることができる。

【0011】

また本発明は、携帯電話機機能と、仮想マシン機能により実行されるアプリケーションプログラムが実装された携帯端末装置が備える制御部に用いるコンピュータで実行される制御用プログラムが、ユーザーの前記携帯端末装置への接触状態を判定する第1の処理と、前記アプリケーションプログラム実行中に、前記第1の処理1によりユーザーの接触状態が解除されたと判定されたときに前記実行中のアプリケーションプログラムの実行を一時中断する第2の処理と、該第2の処理により前記アプリケーションプログラム実行が中断されている状態において、前記第1の処理によりユーザー前記携帯端末装置への接触状態が検知されたと判定されたときに中断中の前記アプリケーションプログラムの実行を再開する第3の処理を含んでいることを特徴としている。

【発明の効果】

【0012】

本発明では、携帯端末装置に、該携帯端末装置がユーザーによって使用されている最中か否かを検出する手段を備え、使用されていないことが検出されたときには、その時点で起動されているJavaやBREWアプリケーションプログラムの実行を自動的に一時中断して端末のバッテリー消費を抑制することができ、バッテリー維持時間の向上を行うことができる。

【0013】

従って、例えば電車内において、携帯電話機でJavaやBREWアプリケーションを実行している途中で電車から降りる際に、携帯電話機をそのまま閉じてしまうなどして、JavaやBREWアプリケーションを終了させずにそのままの状態にバッグやポケットに入れて放置してしまった場合、JavaやBREWアプリケーションが起動し続け、端末のバッテリーを消費して、いざ音声やメールの発信を行おうとしたときに携帯端末が使えない状態となっていたというような問題を解消することができる。

【発明を実施するための最良の形態】

【0014】

図1は、本発明の実施形態の例を示す携帯電話機等の携帯端末装置の概観図である。携帯端末装置11は、キーボードが配置された第1の筐体と表示部が配置された第2の筐体からなり、ヒンジ部により折り畳み可能に構成されている。キーボードが配置された第1の筐体の側面にはタッチセンサ12が配置されており、ユーザーが第1の筐体を手で保持してタッチセンサ12に接触しているときに、それを検出して接触信号を出力する。

【0015】

図2は、図1にて例示した携帯端末装置11の内部構成の概略を示す機能ブロック図である。図2においては、図1の携帯端末装置11を便宜上、携帯端末20と記述する。

【0016】

携帯端末20は、携帯端末20の中に実装されている各デバイスを制御するためのプログラムを実行する制御部21と、携帯端末20にユーザーが触れていることを検知することができるタッチセンサ部22と、携帯端末20上に実装されているキーボタンなどの入力部23と、携帯端末20上に実装されていて制御部21上で動くプログラムから描画や音再生を要求されて表示や音声出力処理する出力部24と、携帯端末20上に実装されているバッテリーなどからの電力処理をする電源部25と、携帯端末20上に実装されてい

10

20

30

40

50

る制御部 2 1 上で動作する制御プログラムからの指示により無線通信を行う無線部 2 6 で構成される。

【 0 0 1 7 】

なお、図 2 では示されていないが、制御部 2 1 は、CPU、ROM、RAM、不揮発性メモリ等を備えており、CPUが、ROMや不揮発性メモリに格納されている制御プログラムあるいはアプリケーションに従って、携帯端末 2 0 の各部を制御している。また、RAMは、ワークエリアとして用いられ、CPUにより実行される種々のアプリケーションや、当該アプリケーションの実行に際して利用される各種のデータ等を一時的に格納する。

【 0 0 1 8 】

ROMには、各種制御情報や制御プログラムの他、例えばCPUにより実行される各種アプリケーション、Javaアプリケーションの実行環境を構築するためのソフトウェア等が格納され、不揮発性メモリには、電話帳、受信した電子メール、IPサーバからダウンロードされたJavaアプリケーション等が格納されている。Javaアプリケーションの実行環境を構築するためのソフトウェアは、CD-ROM等の記憶媒体からインストール、あるいはIPサーバからダウンロードして不揮発性メモリ等に格納してもよい。

【 0 0 1 9 】

図 3 は、携帯端末 2 0 内の制御部 2 1 にて動作するプログラムの構成を示すプログラム構成図である。なお、図 3 に記載するプログラム構成図は、本実施形態を説明するために関連する部分のみを限定して記載したものであり、携帯電話や携帯端末のプログラム構成すべてを記載するものではない。また、このプログラム構成図は、説明の便宜上、Javaアプリケーションを想定した構成となっている。

【 0 0 2 0 】

図 3 に示すプログラム構成図 3 0 において、タッチセンサドライバ 3 1 は、図 2 のタッチセンサ部 2 2 を制御するためのプログラムであり、接触検知処理部 3 2 は、タッチセンサドライバ 3 1 にてタッチセンサ 1 2 へのユーザーの接触を検知した割り込みをあげた際に、接触を検知するプログラムであり、接触状態監視部 3 3 は、接触検知処理部 3 2 にて処理されたユーザーの接触状態を監視して状態を保持するプログラムである。

【 0 0 2 1 】

また、System-AM 3 4 は、システム全体のアプリケーションを管理するアプリケーションマネージメントプログラムであり、Java-AM 3 5 は、Javaアプリケーションのマネージメントプログラムであり、Java-VM 3 6 は、Javaアプリケーションを実行するヴァーチャルマシーンプログラムであり、待ち受け画面表示 APL 3 7 は、携帯電話や携帯端末の電源 ON 後に表示されるデフォルトウィンドウ画面アプリケーションであり、表示出力制御部 3 8 は、Java-AM 3 5 や Java-VM 3 6 や、待ち受け画面 APL 3 7 などからの画面出力要求を実行制御する画面出力制御プログラムである。

【 0 0 2 2 】

図 4 ~ 図 6 は、本実施形態の動作を示すフローチャートである。以下、本実施形態の動作について、図 1 ~ 図 6 を参照して詳細に説明する。

【 0 0 2 3 】

本実施形態の携帯電話および携帯端末において、ユーザーが端末に接触していることを検知する手順は、図 4 に示されているように、まずユーザーの手などがタッチセンサ 1 2 に接触すると、タッチセンサ 1 2 から接触検知信号が出力され、タッチセンサ部 2 2 は、タッチセンサ 1 2 から接触検知信号を受けて、制御部 2 1 に対してユーザーが携帯端末に接触されたことを示す割り込み信号を出力する（ステップ 4 1）。

【 0 0 2 4 】

この割り込み信号を制御部 2 1 が受けると、タッチセンサドライバ 3 1 にて割り込み検知が行われる。タッチセンサドライバ 3 1 は検知された割り込みを基に上位のプログラムに対して、割り込みが検知されたことを知らせる処理を実行する（ステップ 4 2）。

10

20

30

40

50

【 0 0 2 5 】

タッチセンサドライバ 3 1 からユーザーの接触の検知を知らされた接触検知処理部 3 2 は、ユーザーの接触状態を示す状態フラグなどを維持するための処理を行う（ステップ 4 3）。接触状態監視部 3 3 は、接触検知処理部 3 2 のユーザーの接触状態を示す状態フラグの状態を監視しており、上位のプログラムに対して、要求があれば状態の変更があった場合に、状態変化イベントを通知する（ステップ 4 4）。

【 0 0 2 6 】

また、本実施形態における J a v a アプリケーションの起動状況制御手順は、図 5 に示されているように、まず J a v a アプリケーション起動に際して、J a v a - A M 3 5 は、S y s t e m - A M 3 4 に対して、接触検知状態による J a v a アプリケーション起動制御を要求する（ステップ 5 1）。この際に、起動しようとしている J a v a アプリケーションが、ユーザーの接触状態を解除された場合でも起動し続ける対象のアプリケーションかどうかなどを判定して、S y s t e m - A M 3 4 に対して要求を行うようにしてもよい。

10

【 0 0 2 7 】

この J a v a アプリケーション実行中に、ユーザーが携帯端末 1 1 から手を離れた場合、タッチセンサ 1 2 により接触状態の変化が検知され、タッチセンサ部 2 2 は、ユーザーの接触状態が変化したことを制御部 2 1 へ割り込み信号として通知する（ステップ 5 2）。制御部 2 1 にあげられた割り込み信号は、タッチセンサドライバ 3 1 にて割り込みが検知され、タッチセンサドライバ 3 1 は検知された割り込みを処理し、接触検知処理部 3 2 に対して、割り込みが検知されたことを知らせる（ステップ 5 3）。

20

【 0 0 2 8 】

接触検知処理部 3 2 は、ユーザーの接触状態が解除されたことを状態フラグにセットする処理を行う（ステップ 5 4）。接触状態監視部 3 3 は、接触検知処理部 3 2 の状態フラグを確認し、S y s t e m - A M 3 4 に対して、接触状態変化イベントを通知する。この時のイベント内容は、ユーザーの接触が解除された旨を知らせるものとする（ステップ 5 5）。S y s t e m - A M 3 4 は、受け取った接触状態解除イベントに基づき、J a v a アプリケーション起動時に J a v a - A M 3 5 から設定された接触状態解除時の J a v a アプリケーション起動制御方針に従い、J a v a - A M 3 5 に対して、J a v a アプリケーション一時中断処理要求を実施する（ステップ 5 6）。

30

【 0 0 2 9 】

J a v a - A M 3 5 は、S y s t e m - A M 3 4 から指示された J a v a アプリケーション一時中断処理を J a v a - V M 3 6 に対して実施し、J a v a - V M 3 6 は、表示出力制御部 3 8 への描画要求などを停止し、J a v a アプリケーションの実行を停止すると共に、アプリケーションの停止処理完了を、J a v a - A M 3 5 に通知する（ステップ 5 7）。

【 0 0 3 0 】

アプリケーションの停止処理完了通知を受けた J a v a - A M 3 5 は、S y s t e m - A M 3 4 に J a v a アプリケーションの実行を一時停止したことを通知する（ステップ 5 8）。S y s t e m - A M 3 4 は、待ち受け画面表示 A P L 3 7 に対して、起動要求をかけて待ち受け画面表示 A P L 3 7 の起動処理を行ってデフォルトウィンドウ画面を表示させる（ステップ 5 9）。

40

【 0 0 3 1 】

また、本実施形態における一次中断中の J a v a アプリケーションの復帰制御手順は、図 6 に示されているように、まず、J a v a アプリケーション中断中に、ユーザーが携帯端末 1 1 のタッチセンサ 1 2 へ再び触れた場合、タッチセンサ部 2 2 は、ユーザーの接触状態が変化したことを制御部 2 1 へ割り込み通知する（ステップ 6 1）。制御部 2 1 にあげられた割り込みは、タッチセンサドライバ 3 1 にて、割り込み検知され、タッチセンサドライバ 3 1 は検知された割り込みを処理し、接触検知処理部 3 2 に対して、割り込みが検知されたことを知らせる（ステップ 6 2）。

50

【0032】

接触検知処理部32は、ユーザーの接触状態が復帰したことを状態フラグにセットする処理を行う(ステップ63)。接触状態監視部33は、接触検知処理部32の状態フラグを確認し、System-AM34に対して、接触状態変化イベントを通知する。この時のイベント内容は、ユーザーの接触が復帰した旨を知らせるものとする(ステップ64)。System-AM34は、受け取った接触状態復帰イベントに基づき、待ち受け画面表示APL37に対して、デフォルトウィンドウ画面の停止要求を通知する(ステップ65)。

【0033】

待ち受け画面表示APL37は、System-AM34からの停止要求に対して停止処理を実行し、処理が完了したら、System-AM34に処理完了を通知する(ステップ66)。この処理完了を受けたSystem-AM34は、Java-AM35に対して、Javaアプリケーション再開処理要求を実施する(ステップ67)。

10

【0034】

Javaアプリケーション再開処理要求を受けたJava-AM35は、System-AM34から指示されたJavaアプリケーション再開処理をJava-VM36に対して実施し、Java-VM36は、Javaアプリケーションの実行を再開すると共に、アプリケーションの再開処理完了を、Java-AM35に通知する(ステップ68)。Java-AM35は、System-AM34にJavaアプリケーションの実行を再開したことを通知する(ステップ69)。

20

【0035】

このように本実施形態では、携帯端末装置11に設けたタッチセンサ12により、携帯端末がユーザーによって使用されていないことが検出されたときには、その時点で起動されているJavaやBREWアプリケーションプログラムの実行を自動的に一時中断し、その後、タッチセンサ12により、携帯端末がユーザーによって使用再開されたことが検出されたときには、中断中のアプリケーションプログラムの実行を自動的に再開させることができるので、JavaやBREWアプリケーションを実行している途中で、終了させずにそのままの状態にバッグやポケットに入れて放置してしまったような場合であっても、JavaやBREWアプリケーションは自動的に一時中断状態となるので、無駄なバッテリー消費を抑えることができる。

30

【0036】

なお、上記実施形態では、ユーザーによる携帯端末装置の使用状態を検知するために、ユーザーの接触状態を検知するタッチセンサを設けているが、本発明はこの実施形態に限定されるものではなく、非接触状態でも接触状態を擬似的に検知できる静電容量検知等、ユーザーによる携帯端末装置の使用状態を検知可能な適宜の手段を採用することができる。

【0037】

また、携帯端末装置が、ヒンジ部により折り畳み可能に構成された2つの筐体を有している場合には、前記2つの筐体がヒンジ部により開かれた状態と折り畳まれた状態を検知する手段を備え、開かれた状態のときは使用状態と判定し、折り畳まれた状態のときに不使用状態と判定することにより、前記アプリケーションプログラムの動作制御を行うようにしてもよい。

40

【図面の簡単な説明】

【0038】

【図1】本発明の実施形態の例を示す携帯端末装置の概観図である。

【図2】本実施形態の携帯端末装置における内部構成の概略を示す機能ブロック図である。

【図3】本実施形態の携帯端末内の制御部21にて動作するプログラムの構成を示すプログラム構成図である。

【図4】本実施形態において、ユーザーが端末に接触していることを検知する手順を示す

50

フローチャートである。

【図5】本実施形態におけるJavaアプリケーションの起動状況制御手順を示すフローチャートである。

【図6】本実施形態における一次中断中のJavaアプリケーションの復帰制御手順を示すフローチャートである。

【符号の説明】

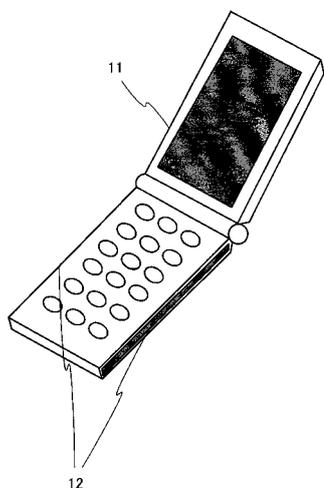
【0039】

- 11 携帯端末装置
- 12 タッチセンサ
- 20 携帯端末
- 21 制御部
- 22 タッチセンサ信号判定部
- 23 入力部
- 24 出力部
- 25 電源部
- 26 無線部
- 30 プログラム構成図
- 31 タッチセンサドライバ
- 32 接触検知処理部
- 33 接触状態監視部
- 34 System-AM
- 35 Java-AM
- 36 Java-VM
- 37 待ち受け画面表示APL
- 38 表示出力制御部

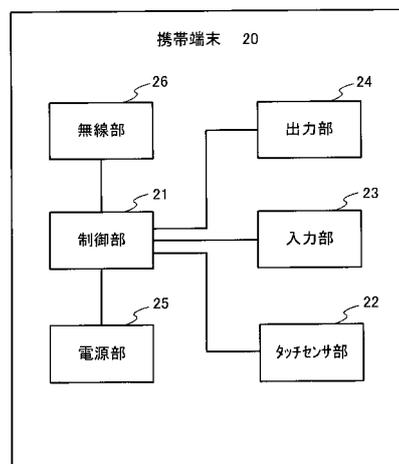
10

20

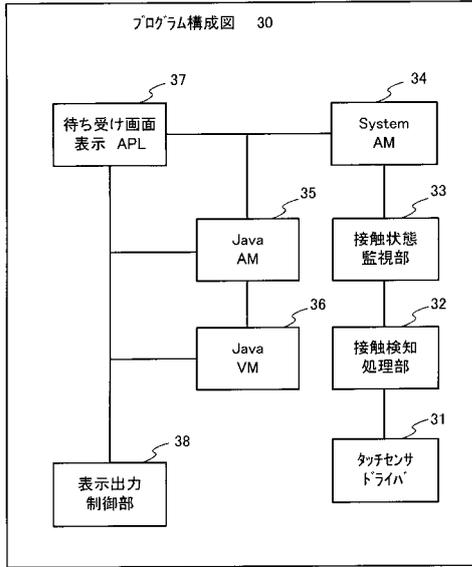
【図1】



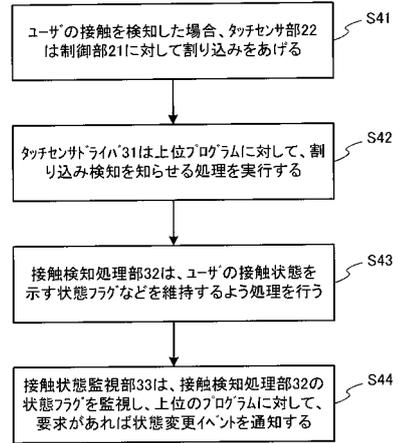
【図2】



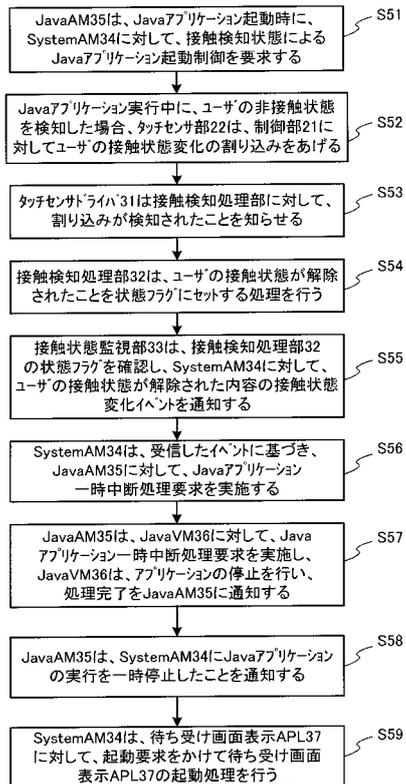
【 図 3 】



【 図 4 】



【 図 5 】



【 図 6 】

