32ビット仮想コンソール

システム仕様

パート8:外部エレメント

資料作成日2023.01.08年Carra著

これは何だ?

このドキュメントは、Vircon32システム仕様のパート8です。この一連のドキュメントは、Vircon32システムを定義し、その機能と動作を詳細に説明する完全な仕様を提供します。

この仕様の主な目的は、Vircon32システムとは何か、およびゲームシステムが準拠していると見なされるために実装される必要がある方法の標準を定義することです。また、Vircon32は仮想システムであるため、これらのドキュメントの重要な2番目の目的は、独自のVircon32実装を作成するための知識を誰にでも提供することです。

Vircon32について

Vircon32プロジェクトはCarraによって独自に作成されました。Vircon32システムとその関連資料(ドキュメント、ソフトウェア、ソースコード、アート、およびその他の関連要素を含む)は、元の作成者が所有しています。

Vircon32は無料のオープンソースプロジェクトで、誰でもゲーム機をプレイし、そのためのソフトウェアを開発できるようにすることを目的としている。この詳細については、使用可能な各ソフトウェアに含まれているライセンステキストを参照してください。

このドキュメントについて

このドキュメントは、Creative Commons Attribution 4.0 License(CC BY 4.0)に基づいて提供されています。ライセンスの全文は、Creative CommonsのWebサイトで読むことができます。

<https://creativecommons.org/licenses/by/4.0/>

**まとめ**

以前の仕様ドキュメントでは、コンソール内のすべての要素が定義されていました。次に、Vircon32システムの一部であり、コンソール自体の外部にあるコンポーネントを説明する必要があります。仕様の第1部では、これらの外部要素の基本的な概要をすでに示しています。次に、第8部では、完全な説明を提供します。

[1 The complete system 3](#_30j0zll)

[2 Gamepads 4](#_1fob9te)

[3 Screen 7](#_3znysh7)

[4 Speakers 8](#_2et92p0)

[5 Cartridges 10](#_tyjcwt)

[6 Memory cards 12](#_3dy6vkm)

## 1完全なシステム

Vircon32システムには、コンソール自体が含まれるだけでなく、コンソールが機能するためには、コンソールの外部にある他のコンポーネントも必要です。完全なVircon32システムは、コンソールと、それに接続されている可能性のあるすべての外部要素で構成されます。これらの要素とその接続は、次の図で確認できます。

コンソールはこのシステムの中心的な要素であり、すべての外部要素はコンソールと排他的に対話すると考えられています。すべての外部要素について、次のセクションでは、その機能と動作、およびコンソールとの通信について説明します。

### 1.1取り外し可能な外部要素

一般的に、すべての外部デバイスは取り外し可能であると考えられています。これらのデバイスが存在しない状態でコンソールの電源を入れることも考えられますが(あまり便利ではありません)、存在するデバイスと存在しないデバイスの組み合わせも考えられます。

これは、完全に準拠するためには、実装がすべての外部デバイスを接続および切断するオプションをサポートする必要があることを意味します。ソフトウェア実装の場合、これはほとんどの場合簡単です。ただし、一部のタイプのハードウェア実装では、これが不可能な場合があります。

これまでのところ、Vircon32ハードウェアはホームコンソールとして描かれていますが、アーケードキャビネットやハンドヘルドデバイスなど、他の形式のハードウェアバージョンも存在する可能性があります。これらの場合、画面、スピーカー、および/またはゲームパッドが固定され、取り外したり切断したりすることができない実装が許容される可能性があります。この例は、次のようなVircon32アーケードキャビネットの概念です:

このようなシステムでは、画面、スピーカー、および4つのゲームパッドがキャビネットに埋め込まれます。つまり、これらは常に存在し、接続されたままになります。システムが完全なシステムのすべての要素をサポートしている限り、これは許容されます。代わりに、このアーケードキャビネットがより薄く、2つのジョイスティックしか含まれていない場合は、2つの追加のゲームパッドを何らかの方法で接続できない限り、完全に準拠しているとは見なされません。

取り外し可能な記憶コンポーネント(カートリッジおよびメモリカード)に関しては、それらは交換可能であることが意図されているので、いかなる実施形態においても取り外し可能であることが要求される。これは、これらの要素の両方がスロットを介して挿入されている画像でも確認できます。

## ゲームパッド2枚

ゲームパッドを使用すると、プレーヤーは指示やボタンを押してゲームを操作できます。コンソールはフレームごとに状態を読み取り、ゲームはこの情報を使用してアクションを実行したり、決定を下したりします。最大4つのゲームパッドをコンソールに接続できます。

Vircon32のゲームパッドは、機能とレイアウトがすべて同じであると見なされます。コンソールは、それぞれが接続されている特定のポートを除いて、それらを区別できません。

### 2.1コントロールとレイアウト

各ゲームパッドには11個のデジタルコントロールがあります。これらは、次のリストに示すように、2つのカテゴリ(4つの方向と7つのボタン)に分割されます:

| 方向:  左、右、上、下 | ボタン:  A、B、X、Y、L、R、開始 |
| --- | --- |

各コントロールには2つの可能な状態しかなく、ゲームパッドはそれらをそれぞれ1ビットで表し、1=押された状態と0=押されていない状態です。すべてのコントロールのニュートラル状態は押されていません。各コントロールは、ユーザーが力を加えるなどのアクティブなアクションを実行した場合にのみ押された状態になります。

Vircon32ゲームパッドは、コントロールを物理的に別々のグループに編成しています:方向パッド(左、右、上、下)、4つの前面ボタン(A、B、X、Y)、2つの肩ボタン(L、R)、および中央ボタン(スタート)。すべての実装で、各ゲームパッドのコントロールは、この図に示されているレイアウトと配置に従う必要があります。

これは、ゲームクリエイターがコントロールスキームを実装する方法を決定するときに、このコントロールレイアウトを安全に想定できることを意味する。例えば、シューティングゲームでは、4つの前面ボタンのダイヤモンド形状を想定して、それらを「2次方向パッド」として機能させ、どの方向にシュートするか(Xは上に、Yは左にシュートするなど)を制御することができる。

Vircon32ゲームパッドには、ゲームパッドコントロールのための追加のメカニズムや自動化は含まれていません。これは、ターボファイア、オートファイア、オートホールド、またはいかなる種類のプログラミング可能性もないことを意味します。

### 2.2方向パッド(dパッド)

方向性パッドはさまざまな方法で実装でき、実装ごとに選択できます。可能なオプションは、4つの個別のボタン、十字形、明示的な対角線用の八角形などです。選択されたオプションの出力は、常に4ビット(4つの方向のそれぞれに1つ)である必要があります。

すべてのVircon32 Dパッドは、反対方向が同時に押されることを防ぐためのメカニズムを実装する必要があります。これは、コンソールが常に一貫したDパッド方向を解釈できるようにするために必要です。これは、次のことを意味します:

* Dパッドの水平軸は、左と右の両方を同時に押さないようにする必要があります。
* Dパッドの垂直軸は、UpとDownの両方を同時に押さないようにする必要があります。

これは、さまざまなソリューションを使用して実装できます。例として、従来のコンソールでは通常、傾斜機構が使用されていました。この簡単な図を次の図に示します。同じ軸の両端を考慮すると、ここで呼ばれる**[-]**および**[+]キー**すると、次のようになります。

### 2.3コンソールへの接続

ゲームパッドからコンソールへの通信は、完全に単一方向であると考えられています。コンソールは、ゲームパッドの動作や状態に影響を与えることはできません。コンソールは、各ポートを介してゲームパッドの状態を読み取るだけです。

実装では、任意の配線とコネクタ、および電気信号とデータ送信のための任意のプロトコルを選択できます。ただし、選択されたコネクタには、ゲームパッドが4つのポートのそれぞれに接続されているかどうかをコンソールがいつでも知ることができるようにする何らかの種類の信号またはセンサーが含まれている必要があります。この情報は1ビットで表されます。1=接続、0=切断です。

## 3画面

この画面は、コンソールによって生成された画像を表示し、プレーヤーがそれを見ることができるようにするために使用される。この画面は、GPUによってフレームごとに送信されるピクセルカラー情報を受信し、それを使用して必要な画像を生成する。

### 3.1スクリーン寸法

Vircon32の画面は、水平方向に640ピクセル、垂直方向に360ピクセル(合計230400ピクセル)の矩形マトリックスです。ピクセルは正方形(1:1のアスペクト比)で、その間にギャップがないため、全画面のグローバルアスペクト比は16:9になります。表示サイズは目的に関係なく、実装ごとに自由に選択できます。

画面のピクセルは、左上(0,0)から右下(639,359)までの番号が付けられていると見なされます。画面には、異なるビデオモードやディスプレイ設定はありません。イメージプロパティは、すべての実装のすべてのゲームで同じです。

Vircon32の画面は、常に固定された水平位置で表示されると想定されています。実装にはこの機能が必要ないため、ゲーム作成者は画面を回転できるとは想定できません。これは、たとえば、Vircon32のゲームでは、垂直方向のイメージ比率を9:16にするために、ユーザーが画面を90度回転する必要がないことを意味します。

### 3.2画質

画面には、24ビット(True Color)の範囲のRGB空間の色が表示されます。各ピクセルには、赤、緑、青の3つのチャネルがあります。各チャネルの色深度は8ビットで、0(純粋な黒)から255(フル強度)まであります。画面は、各ピクセルに対して受信された色でのみビデオを生成します。最終的なイメージには、色のブレンドやフィルタリングはありません。

画面には毎秒60フレームが表示されます。常にフルフレームを使用してビデオを生成します。補間やインターレース効果はありません。実際の画面には常に特定の入力ラグがあるため、実装ではこれを管理する方法を決定する必要があります。

### 3.3コンソールへの接続

コンソールから画面への通信は、完全に単一方向であると考えられています。画面は、コンソールの動作や状態に影響を与えることはできません。コンソールは、コネクタを介してビデオ信号を送信するだけで、接続されている画面があるかどうかを判断することもできません。

実装では、任意の配線とコネクタ、および電気信号とデータ送信用の任意のプロトコルを選択できます。ただし、選択された通信には、各フレームの正しいタイミングを保証する方法が含まれている必要があります。

GPUおよびSPUドキュメントで説明されているように、実際には、ディスプレイにオーディオ機能を統合することも一般的です。HDMIなどの現在の通信プロトコルでは、オーディオとビデオを同じ信号で結合できるため、配線とコネクタを共有できます。したがって、画面とスピーカーが別々のエンティティとして扱われている場合でも、オーディオとビデオの出力を結合して1つのデバイスに送信される結合信号にする実装が有効です。

## 4スピーカ

スピーカーは、コンソールによって生成されたサウンドを再生し、プレーヤーに聞こえるようにするために使用されます。スピーカーは、SPUによって送信されたサンプルの連続シーケンスを受信し、それを使用して必要なサウンドを生成します。

### 4.1サウンドチャンネル

Vircon32は、2.0個のスピーカーの標準セットを使用してステレオサウンドを生成します。これは、プレーヤーに対して異なる配置の2つの独立した音源があることを意味します。1つは左側に、もう1つは右側にあります。

この文書では、各音源を別々のスピーカーとして扱いますが、1つのデバイスが左右の2つの音源を統合する実装も可能です。

### 4.2音質

サウンドは44100 Hzのサンプリングレートで生成され、サンプルは16ビットの精度を持ちます。各スピーカーは、SPUによって送信された一定の個別のサンプルシーケンスを受信し、それを使用してそのチャネル(LeftまたはRight)に必要なサウンドを生成します。

ヘッドフォンはスピーカーとして使用できますが、4人のプレーヤー全員が聞くことのできる音源が常に存在する必要があります。どのプレーヤーが聞いた音とも、音質、ラグ、歪みに違いがないようにしてください。

Vircon32スピーカーは、シーケンス内の各サンプルに対して受信した値のみを使用してサウンドを生成します。最終的なサウンドには、フィルタやイコライザの調整はありません。

### 4.3コンソールへの接続

コンソールからスピーカーへの通信は、完全に単方向であると考えられています。スピーカーはコンソールの動作や状態に影響を与えることはできません。コンソールはオーディオ信号をコネクタを介して送信するだけであり、接続されているスピーカーのセットがあるかどうかを判断する方法もありません。

実装は、任意の配線およびコネクタ、ならびに電気信号およびデータ伝送のための任意のプロトコルを選択することができる。しかし、選択された通信は、左および右スピーカーのためのシーケンスを分離する方法を含み、各サンプルのための正しいタイミングを確保する必要がある。

スクリーンに関して既に述べたように、スクリーンおよびスピーカが別個のエンティティとして扱われる場合であっても、オーディオおよびビデオ出力を単一のデバイスに送信される組み合わされた信号に結合することは、実装にとって有効である。

## 5カートリッジ

カートリッジは、Vircon32の主要なストレージサポートです。一般的に、各カートリッジは個別のゲームを格納し、コンソールはカートリッジからそのプログラムを実行します。カートリッジはそれぞれ異なる内容を持つことができますが、それらはすべて同じ構造とコネクタを持ち、それらを交換可能にします。

### 5.1カートリッジの構造

Vircon32カートリッジには、最大3つの独立した読み取り専用メモリ(ROM)(プログラム、ビデオ、オーディオ)を搭載できます。コンソールからの通信は、すべてのカートリッジROMに関する情報を保持し、その内容へのアクセスを管理する何らかのコネクタを介して行われます。

すべてのカートリッジにはプログラムROMが必要ですが、オーディオおよびビデオROMは両方ともオプションです。3つのROMのそれぞれは、存在する場合、異なるカートリッジ内に可変数の要素(単語/画像/音)を有することができる。

コンソールは、(実装によって選択された)何らかの方法で各カートリッジの実際の内容を決定する必要があります。これは、カートリッジに追加情報(ヘッダーまたはメタデータ)を含めるか、コンソールが特定のカートリッジごとに内部構造を決定できるようにコンソールとカートリッジの通信を設計することによって行うことができます。

3つのROMはすべて、圧縮を使用せずに内容を保存します。それぞれの容量制限は次のとおりです。

* プログラムROMには、最大128 MWords=128 x 1024 x 1024ワード=512 MBを格納できます。
* ビデオROMには、最大256個のテクスチャ=256 x 1024 x 1024ピクセル=1 GBを格納できます。
* オーディオROMには、最大1024個のサウンドと256 x 1024 x 1024サンプル=1 GBを格納できます。

これは、カートリッジの最大可能ストレージが合計2.5 GBであることを意味します。

### 5.2メモリのパフォーマンス

具体的なパフォーマンスレベルは示されていませんが、3つのカートリッジROMは、次の機能を実行するのに十分な速度であると想定されています。

* サイクルごとに、プログラムROMメモリは、CPUが要求するすべてのワードを、その同じサイクル内で現在の命令を完了するのに間に合うようにCPUに提供することができる。
* すべてのフレーム、ビデオROMメモリは、同じフレーム内のすべての描画操作を完了するために必要なすべてのピクセルをGPUに提供できます。
* すべてのフレーム、オーディオROMメモリは、SPUがその同じフレーム内のすべてのサウンド生成を完了するために必要なすべてのサンプルをSPUに提供することができる。

### 5.3コンソールへの接続

カートリッジからコンソールへの通信は、完全に単一方向であると考えられています。カートリッジはパッシブコンポーネントであり、すべてのメモリが読み取り専用であるため、コンソールはその内容を読み取ることはできますが、その状態に影響を与えることはできません。

実装は、任意の配線およびコネクタ、ならびに電気信号およびデータ伝送のための任意のプロトコルを選択することができる。ただし、選択されたコネクタおよび/またはプロトコルが満たさなければならないいくつかの要件がある:

* それらは、カートリッジがそのカートリッジスロットに接続されているかどうかをコンソールがいつでも知ることができるようにする何らかの種類の信号またはセンサーを含む必要があります。この情報はシングルビットで表されます。1=接続、0=切断です。
* 選択された通信は、それらがすべて交換可能であることを保証するために、その特定の実装のために設計されたすべてのカートリッジに対して同じである必要がある。

## メモリカード×6

メモリカードは、コンソールに小さな永続的なストレージを提供し、ゲームはこれを使用して異なるセッション間でデータをロードおよび保存できます。カートリッジとは異なり、すべてのVircon32メモリカードはすべて同一と見なされます。コンソールがメモリカードを区別する唯一の方法は、その内容を調べることです。

### 6.1メモリカードの構造

すべてのVircon32メモリカードは機能的に同一であり、すべて同じ構造と記憶容量を持っています。メモリカードには内部ランダムアクセスメモリ(RAM)があり、それに含まれる任意の単語を個別に読み書きできます。コンソールとの通信は、内部メモリへのアクセスを管理する何らかのコネクタを介して行われます。

コンソールRAMとは異なり、メモリカードRAMはその内容を永続的に保存します。これらは電源オフ/リセット時や、カードがコンソールから取り出されたときには削除されません。Vircon32メモリカードは圧縮を使用せずに内容を保存し、すべての固定容量は256 KWords=256 x 1024ワード=1 MBです。

### 6.2メモリのパフォーマンス

特定の性能レベルは述べられないが、カードメモリは、すべてのサイクルが、その同じサイクル内で現在の命令を完了するのに間に合うように、CPUのすべての要求に応答できるように、十分に高速であると仮定される。

### 6.3コンソールへの接続

メモリーカードからコンソールへの通信は双方向です。メモリーカードは受動コンポーネントですが(したがって、通信は常にコンソールからトリガーされます)、カードメモリーは読み取りと書き込みが可能です。したがって、カートリッジとは異なり、コンソールはメモリーカードの内容を変更することができます。

実装では、任意の配線とコネクタ、および電気信号とデータ伝送のための任意のプロトコルを選択できます。ただし、選択されたコネクタには、メモリカードがカードスロットに接続されているかどうかをコンソールがいつでも知ることができるようにする何らかの信号またはセンサーが含まれている必要があります。この情報は1ビットで表されます:1=接続、0=切断。また、選択された通信は、すべての互換性を確保するために、その特定の実装のために設計されたすべてのメモリカードで同じである必要があります。

(パート8の終わり)