**1.はじめに**

IPMI（Intelligent Platform Management Interface）とはサーバのCPU、バス、ファン、温度センサ、電源などの基本コンポーネントの監視や遠隔制御などを行うためのインターフェースである。IPMIに対応したサーバは、OSやアプリケーションソフトなどを介さずに、ネットワークを通じて管理用端末のシステム監視ソフトなどと直接通信することができ、遠隔からシステムを監視や制御することができる。しかし、市販のIPMIカードは特定のサーバマザーボードにしか対応しない。そこで、IPMIカードが一般のマザーボードにも対応可能であれば、一般のマザーボードの利便性が向上につながると考えられる。

本研究では一般のマザーボードに対応できるIPMIモジュールを構築する。

**2.システム構成**

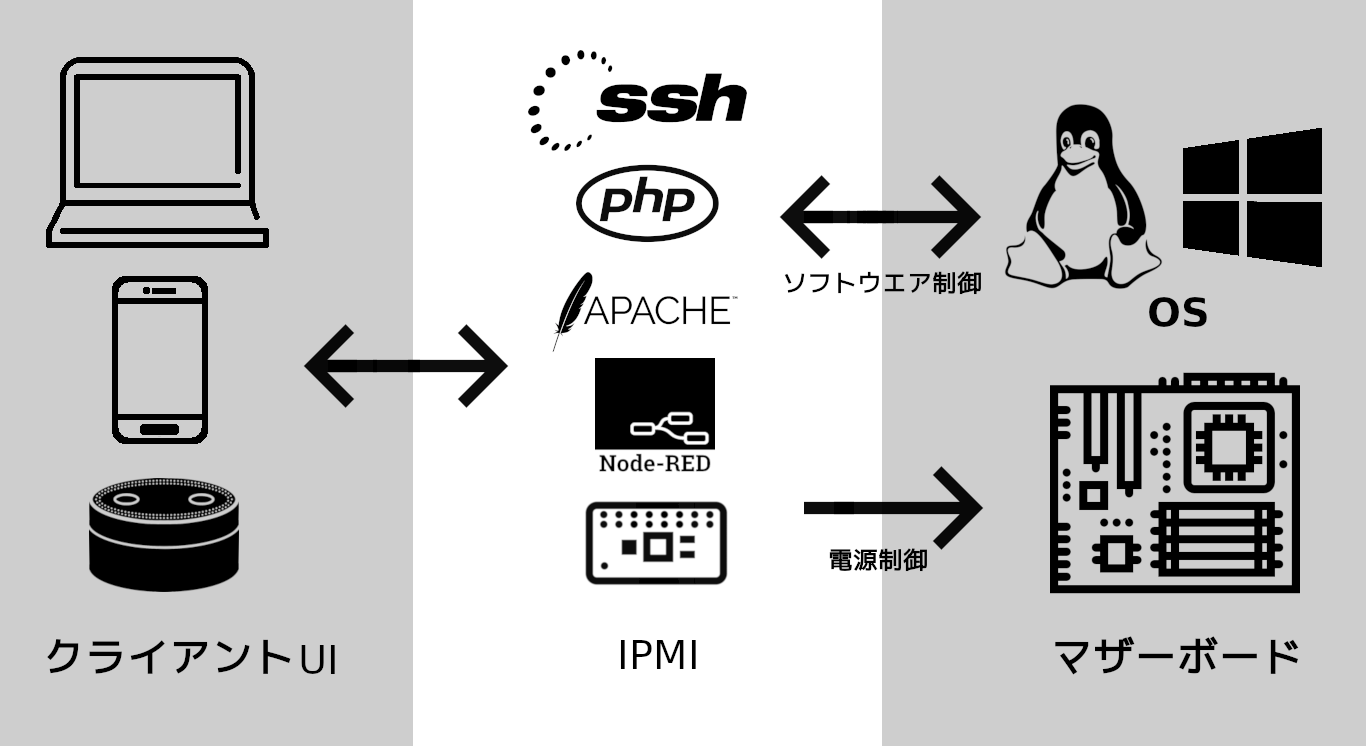


図１．システム構成図

図1にシステム構成図を示す。各部の動作を説明する。

* クライアントUI

クライアントUIはスマホ、パソコンなどの端末でSSH、又はHTTPプロトコルによってIPMIモジュールとデータをやり取りする。データの内容はホストPCの電源状態、IPMIモジュールとの接続状態、ホストPCに実行させたい命令、命令実行の結果である。

* IPMI

IPMIはUSBネットワークインタフェースとしてホストPCに接続したRaspberry Pi Zero Wを用いる。IPMIモジュールはホストPCと独立の電源を使用し、マザーボードの電源スイッチを物理的に操作することによって、ホストPCの電源状態と関係なく制御することができる。また、IPMIモジュールはUSBネットワークインタフェースとしてホストPCに接続することで、ホストPCがネットワークに接続しない状態でも遠隔操作ができる。

Apache+Phpウエブサーバーでクライアント端末にウエブインタフェースを提供する。スマートスピーカに関する処理はnoderedで行う。使用可能なインタフェースはSSH、ウエブである。

* ホストPC

ホストPCはSSHプロトコルでIPMIモジュールからの命令を受取り、命令の実行結果をIPMIモジュールに送る。

**3.電源制御**

**3.1電源スイッチ制御する回路**

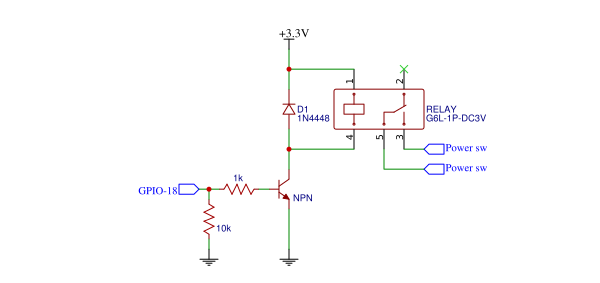


図2.電源スイッチ制御する回路図

図2は電源スイッチ制御する回路図を示す。ホストPCの電源がOFFの状態でも起動させるには物理的にマザーボードの電源スイッチを操作することが必要である。そこで電源スイッチにリレーを並列接続して用いる。Raspberry Pi Zero W　GPIOの出力電流が小さく、直接リレーを制御することができないため、NPNトランジスタスイッチ回路を用いてリレーを制御する。

**3.2電源状態監視する回路**

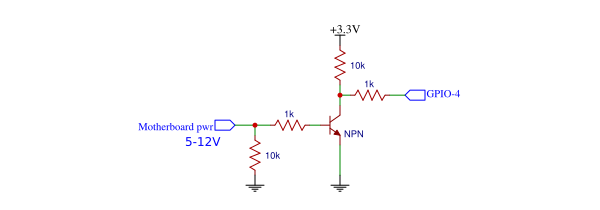


図3.電源状態監視する回路図

　図3は電源状態監視する回路図を示す。PCが稼動していないときにも5Vバスに電源が供給される場合があるため、5Vバスでホストの電源状態を判定できないときは12VバスでホストPCの電源状態を判定する。Raspberry Pi Zero Wの最大入力可能電圧は3.3Vであるため直接マザーボードの5V、または12Vバスに直接接続せず、NPNトランジスタスイッチ回路でホストPCの電源状態を判定する。

**4.コマンドラインインターフェース**

ホストPCを遠隔操作するためにSSHプロトコルを利用する。IPMIモジュールにインストールされたIPMIツールでホストPCの電源やソフトウエア制御を行う。

使用可能なコマンドラインを以下で示す。

* start/stop/restart : ホストPCを起動/シャットダウン/再起動する。
* -f ： シャットダウン、再起動の命令を実行後、一定の時間たってもホストPCの電源状態が変化しないとき強制的に電源を切る。
* status : ホストPCの状態を表示する。
* setup : IPMIモジュールの設定を行う。ホストPCをアクセスためのIPアドレスやwebインターフェースアクセスためのパスワードを変更することができる。
* shell : ホストPCのソフトウェア制御を行う。ホストPCがWindowsの場合はpowershell、Linuxの場合はbash/zshなどに接続する。ホストPCのパスワードが設定された場合、powershell/bashに接続する前にホストPCのユーザー名やパスワードの入力を行う。セキュリティ上のため、ホストPCのユーザー名やパスワードなどはIPMIモジュールに保存しない。
* help : IPMIのマニュアルを表示する。
* version : IPMIのソフトウェアバージョンを表示する。

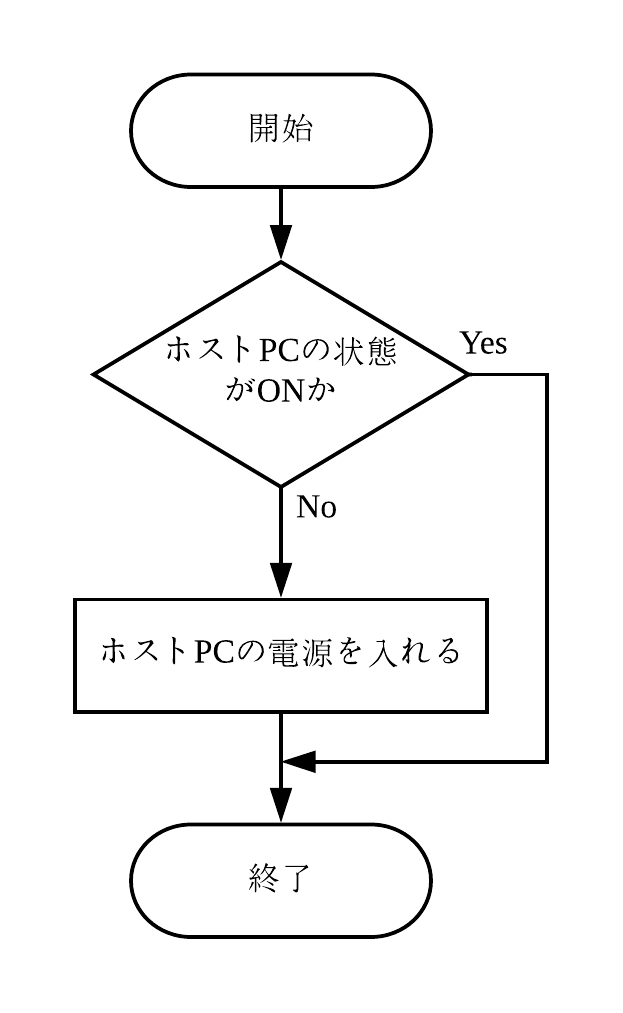


図4.startコマンドのフローチャート

　図4はホストパソコンを起動するためのstartコマンドのフローチャートを示す。startコマンドが実行されたとき、ホストPCがすでに起動していた場合、プログラムを終了する。ホストPCの電源状態がOFFの場合、起動命令を実行する。

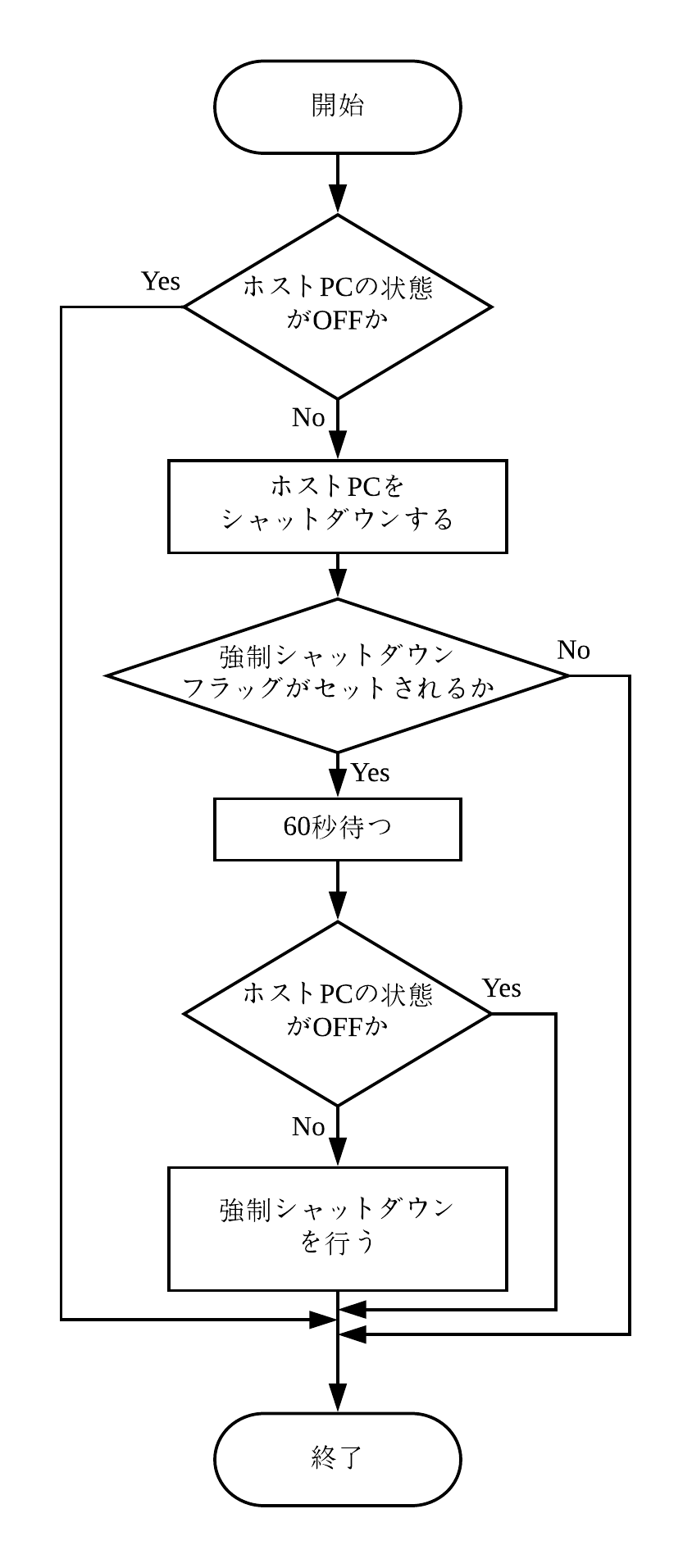


図5.stopコマンドのフローチャート

図5はホストパソコンをシャットダウンするためのstopコマンドのフローチャートを示す。stopコマンドが実行されたとき、ホストPCの電源状態がOFFの場合、プログラムを終了する。ホストPCの電源状態がONの場合、シャットダウン命令を実行する。強制シャットダウンフラッグ（-f）が設定された場合、シャットダウン命令を実行後、一定の時間たってもホストPCの電源状態が変化しないとき強制的に電源を切る。

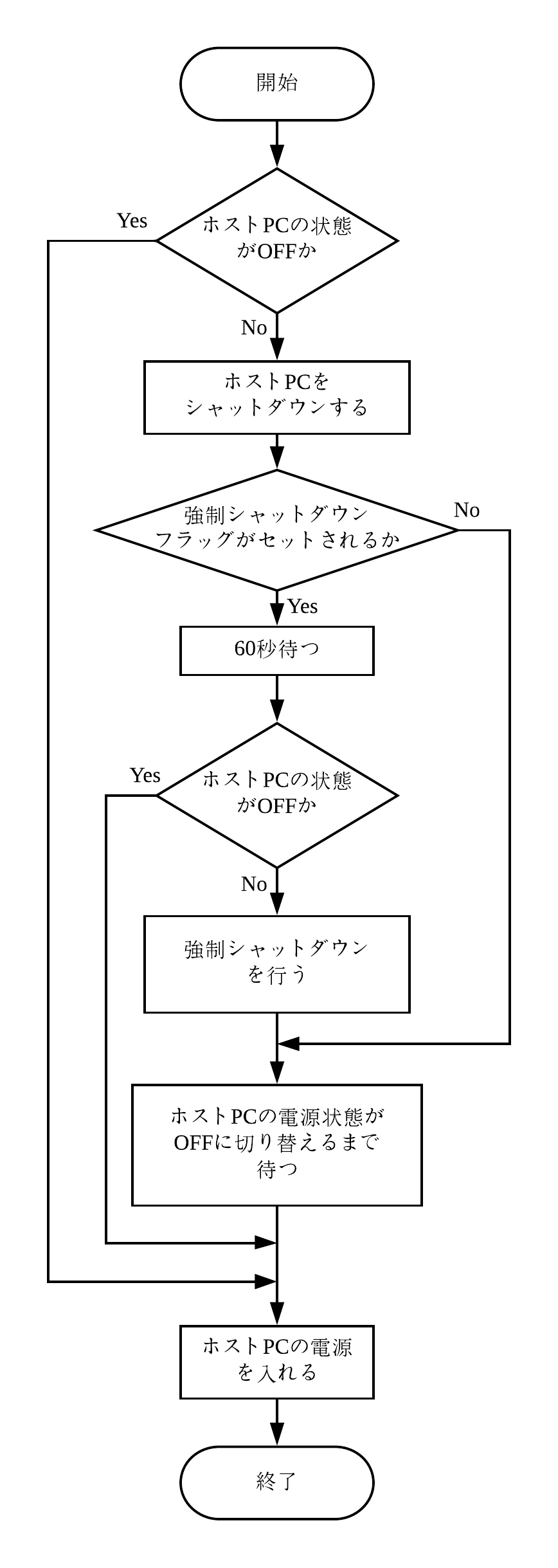


図6.Restartコマンドのフローチャート

　図6はホストパソコンを再起動するためのrestartコマンドのフローチャートを示す。restartコマンドは2段階で実行される。第1段階はシャットダウン命令を実行する。ホストPCの電源状態がOFFに切り替わった後、第2段階の起動命令を実行する。

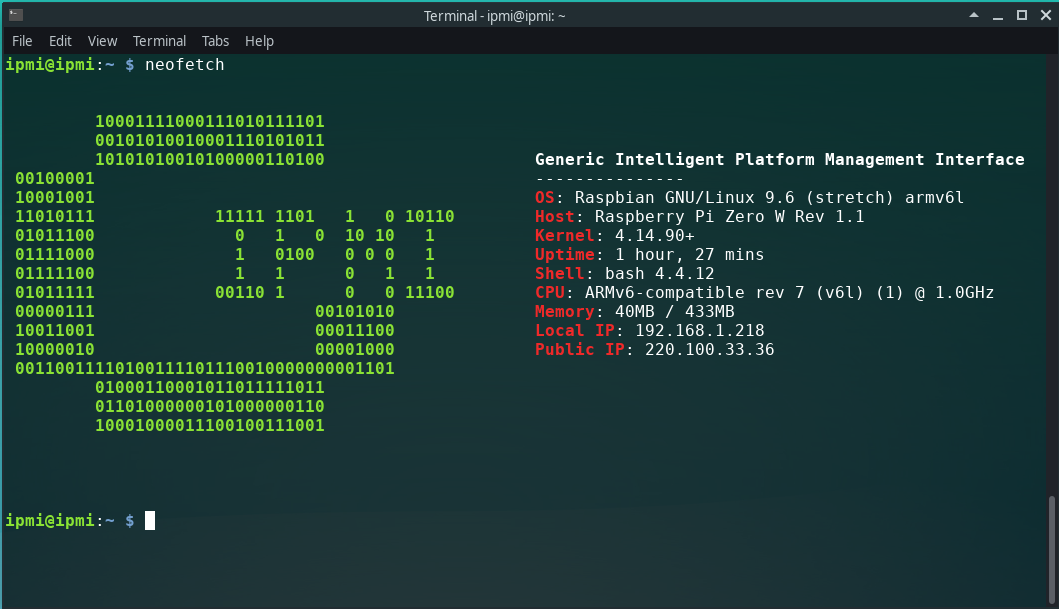


図7.パソコンのコマンドラインインターフェース



図8.スマートフォンのコマンドラインインターフェース（ssh juiceアプリを利用）

　図7はパソコンのコマンドラインインターフェースを示す。図8はssh juiceアプリを利用するときのスマートフォンのコマンドラインインターフェースを示す。

**5.ウェブインターフェース**

ウェブインインターフェースでホストPCの監視や電源制御を行うことができる。ウェブインインターフェースはBootstrapフレームワークを利用することでアクセス端末の画面サイズによって適切なレイアウトを提供する。図9はパソコンのウェブインインターフェースを示す、図10はスマートフォンのウェブインインターフェースを示す。

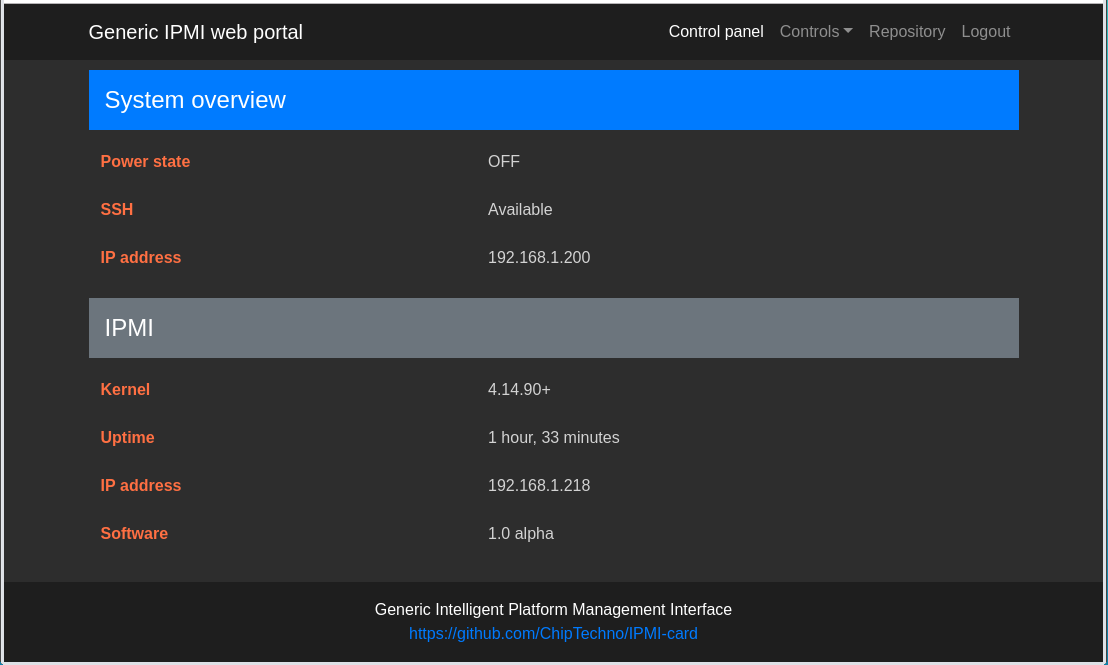


図9.パソコンのウェブインインターフェース

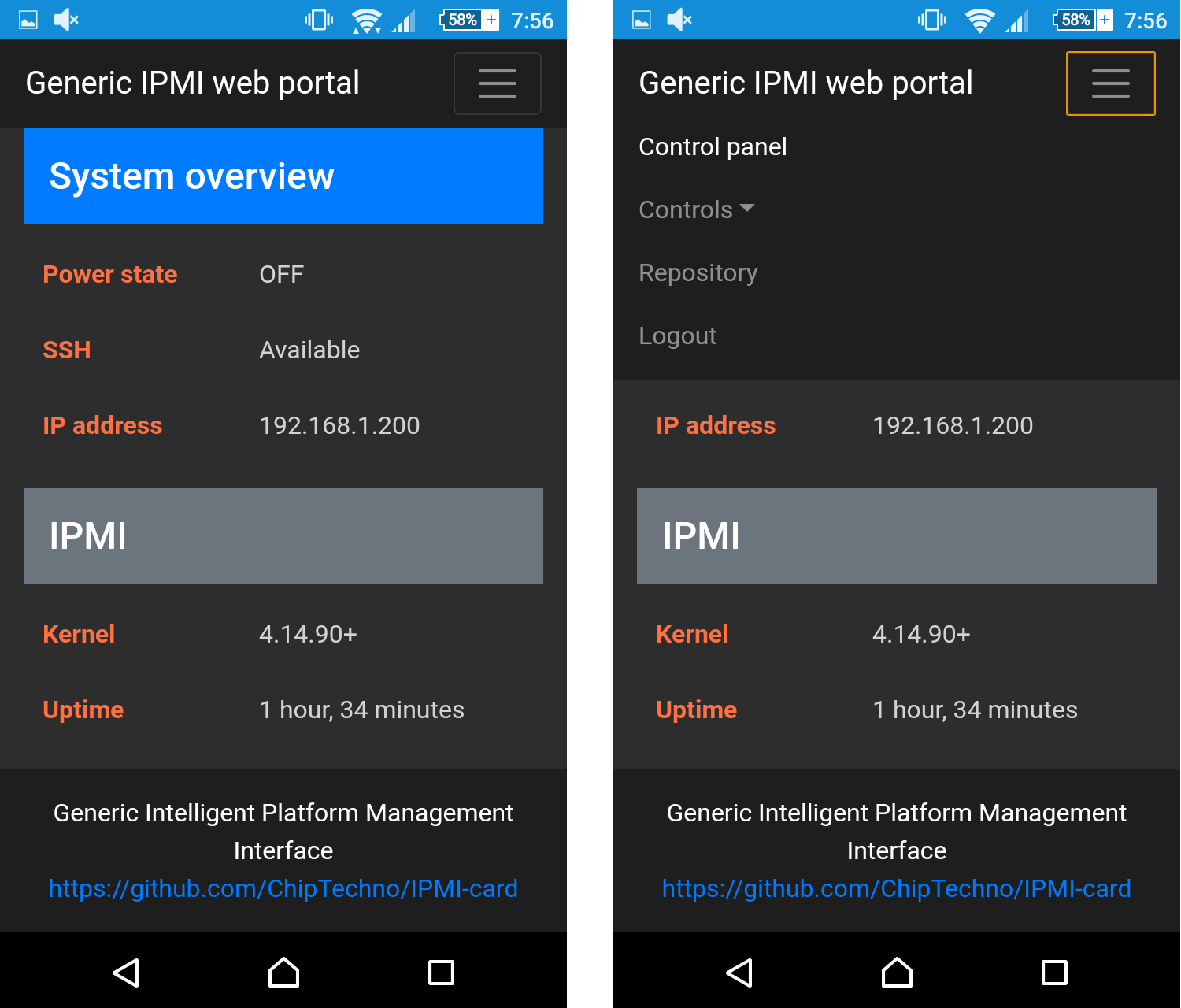


図10.スマートフォンのウェブインインターフェース

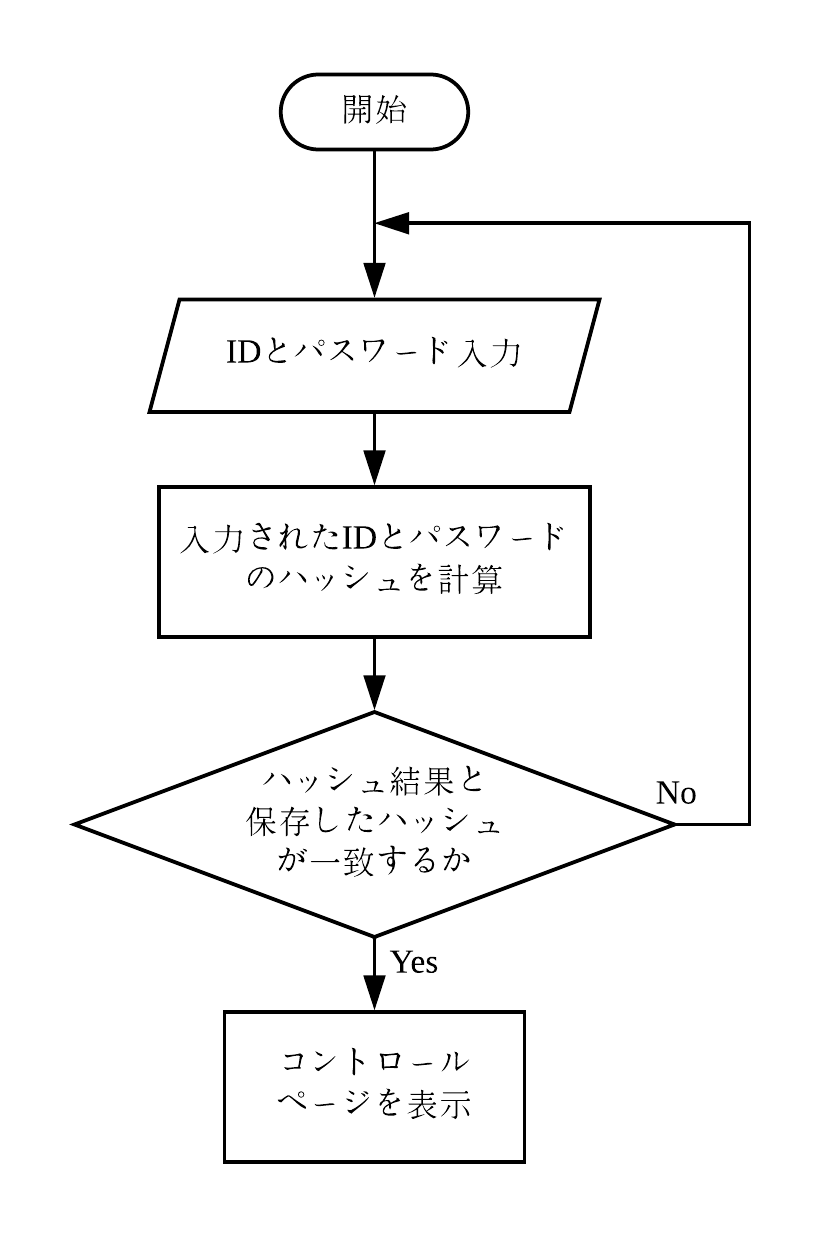


図11.ウェブインインターフェース認証機能のフローチャート

図11はウェブインインターフェース認証機能のフローチャートを示す。ウェブインインターフェースをアクセスためにCLIと別のIDやパスワードを入力することが必要である。ウェブインインターフェースのIDやパスワードの設定はIPMIツールのsetup機能で設定することができる。設定されたIDとパスワードはSHA256のハッシュ関数で暗号化しIPMIモジュールの設定ファイルに保存する。

**6.スマートスピーカーインターフェース**

スマートスピーカーインインターフェースはnoderedを利用し、スマートスピーカーとの連携を行う。スマートスピーカーの音声認識機能でホストPCの電源制御を行うことができる。スマートスピーカーは命令入力に適さないため、使用可能な機能は電源制御のみとなる。

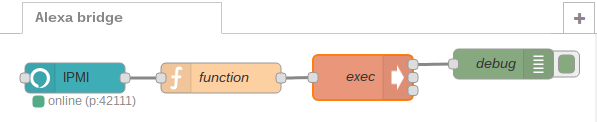


図12.NoderedによるAmazon echoとの連携

　図12はnoderedによるAmazon echoとの連携を示す。NoderedとAmazon echoを連携するためにnode-red-contrib-alexa-localノードを利用し、IPMIモジュールを一般のスマートホームデバイスとしてAmazon echoと連携する。

**7.動作テスト**

**7.1電源制御機能**

表1.電源制御機能のテストケース

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 番号 | テスト内容 | 想定結果 | 判定 | その他 |
| 1 | ホストPCの電源がOFF時、startコマンドを実行する。 | ホストPCを起動する。 | ○ |  |
| 2 | ホストPCの電源がON時、startコマンドを実行する。 | ホストPCの電源状態が変わらない。 | ○ |  |
| 3 | ホストPCの電源がOFF時、stopコマンドを実行する。 | ホストPCの電源状態が変わらない。 | ○ |  |
| 4 | ホストPCの電源がON時、stopコマンドを実行する。 | ホストPCをシャットダウンする。 | ○ |  |
| 5 | ホストPCの電源がOFF時、stop -fコマンドを実行する。 | ホストPCの電源状態が変わらない。 | ○ |  |
| 6 | ホストPCの電源がON時、stop -fコマンドを実行する。 | シャットダウン命令を実行した後、一定の時間（60秒）たってもホストPCの電源状態が変化しないとき強制的に電源を切る | ○ |  |
| 7 | ホストPCの電源がOFF時、startコマンドを実行する。 | ホストPCを起動する。 | ○ |  |
| 8 | ホストPCの電源がON時、restartコマンドを実行する。 | ホストPCを再起動する。 | ○ |  |
| 9 | ホストPCの電源がOFF時、restart -fコマンドを実行する。 | ホストPCを起動する。 | ○ |  |
| 10 | ホストPCの電源がON時、restart -fコマンドを実行する。 | シャットダウン命令を実行した後、一定の時間（60秒）たってもホストPCの電源状態が変化しないとき強制的に電源を切って、再起動する。 | ○ |  |
| 11 | ホストPCの電源がOFF時、statusコマンドを実行する。 | 「Power state : OFF」を表示する。 | ○ |  |
| 12 | ホストPCの電源がON時、statusコマンドを実行する。 | 「Power state : ON」を表示する。 | ○ |  |
| 13 | ウェブインインターフェースでstartコマンドを実行する。 | Phpのshell\_exec関数でipmi startコマンドを実行する。 | ○ |  |
| 14 | ウェブインインターフェースでstopコマンドを実行する。 | Phpのshell\_exec関数でipmi stopコマンドを実行する。 | ○ |  |
| 15 | ウェブインインターフェースでrestartコマンドを実行する。 | Phpのshell\_exec関数でipmi restartコマンドを実行する。 | ○ |  |
| 16 | スマートスピーカーでパソコンを起動する。 | Noderedでipmi startコマンドを実行する。 | ○ |  |
| 17 | スマートスピーカーをシャットダウンする。 | Noderedでipmi stopコマンドを実行する。 | ○ |  |

　表1は2019年1月7日に実行した電源制御機能のテストケース結果を示す。電源制御機能が正常に動作することがわかった。

**7.2ウェブインインターフェース認証機能**

表2.ウェブインインターフェース認証機能のテストケース

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 番号 | テスト内容 | 想定結果 | 判定 | その他 |
| 1 | ログインしない状態でホームページ（index.php）をアクセスする | ログインページに移動する。 | ○ |  |
| 2 | ログインした状態でホームページ（index.php）をアクセスする | コントロールページに移動する。 | ○ |  |
| 3 | ログインしない状態でホームページ以外のページにアクセスする | ログインページに移動する。 | ○ |  |
| 4 | ログインした状態でホームページ以外のページにアクセスする | アクセスしたページを表示する。 | ○ |  |
| 5 | ログインページで正しいIDやパスワードを入力 | コントロールページに移動する。 | ○ |  |
| 6 | ログインページで誤りIDまたはパスワードを入力 | エラーメッセージを表示し、ログインに移動する。 | ○ |  |
| 7 | ログインページでインジェクション攻撃を行う | インジェクションされた命令を実行せず、エラーメッセージを表示し、ログインに移動する。 | ○ |  |
| 8 | ログアウトする | 記録したsession keyを消し、ログインページに移動する。 | ○ |  |

表2は2019年1月7日に実行したウェブインインターフェース認証機能のテストケース結果を示す。ウェブインインターフェース認証機能が正常に動作することがわかった。

**7.3ペネトレーション・脆弱性検知テスト**

ペネトレーションテストはroutersploitツールを使い、IPMIモジュールの脆弱性テストを行った。2019年1月8日の実行は「脆弱性が発見されなかった」という結果を得た。

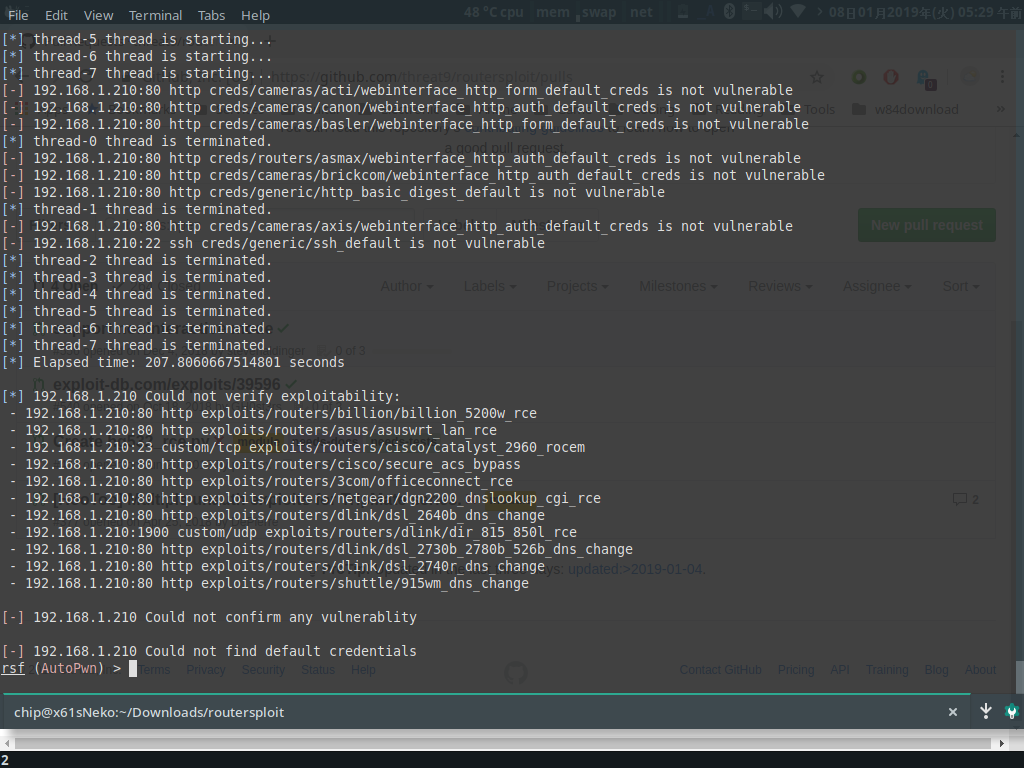


図13.ペネトレーションテスト結果

図13は2019年1月8日に実行したペネトレーション・脆弱性テスト結果を示す。脆弱性が発見されなかったということが分かった。

**7.4消費電力測定**

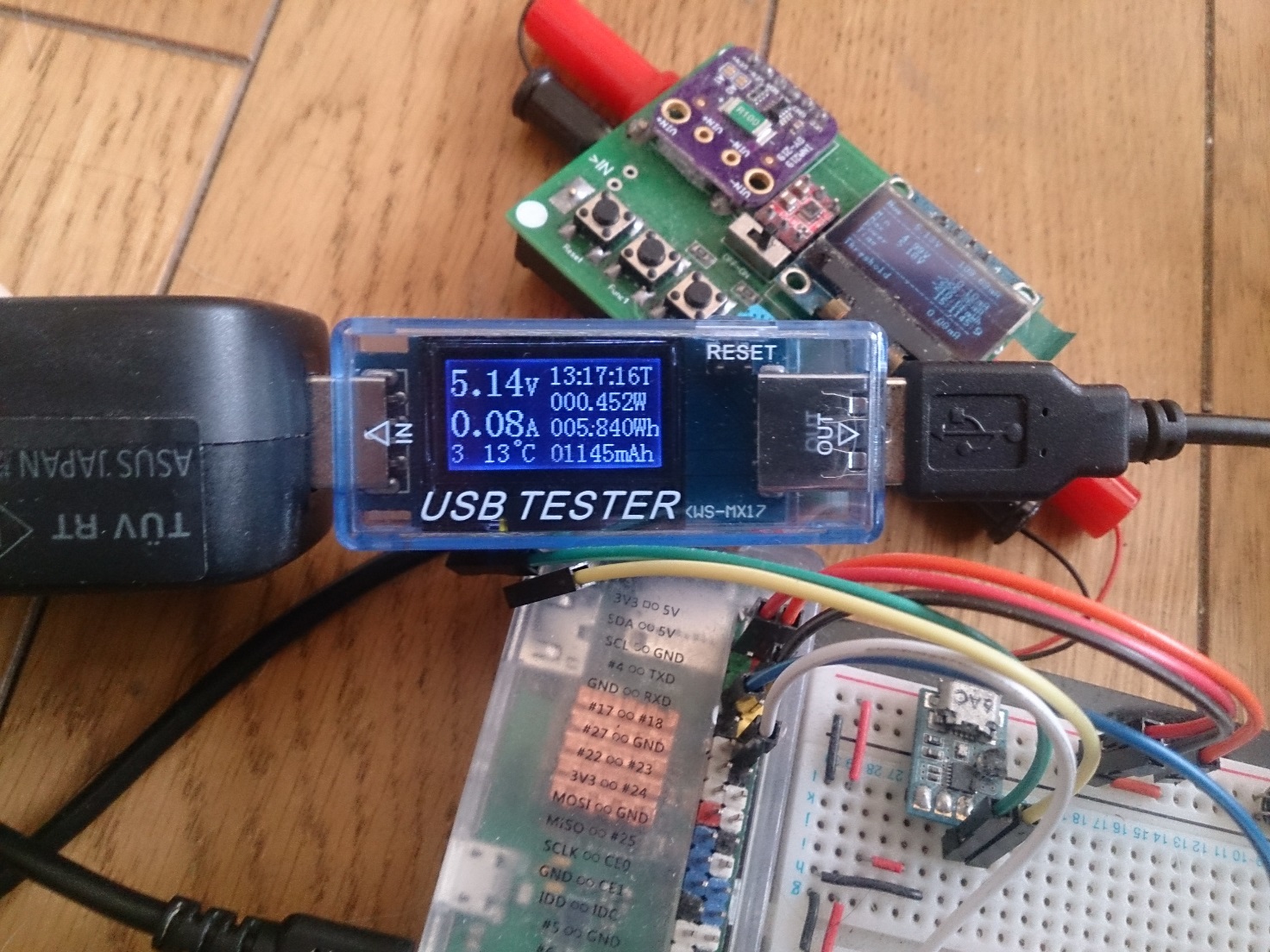


図14.消費電力測定結果

　図14は2019年1月21日に消費電力測定結果を示す。測定時間は13時17分16秒間で合計消費電力は5.840Whという結果を得た。

表3.時間平均消費電力測定結果

|  |  |  |
| --- | --- | --- |
| 測定時間 | 消費電力(Wh) | 時間平均消費電力(W) |
| 13時17分16秒間 | 5.840 | 0.439 |

　表3は時間平均消費電力測定結果を示す。時間平均消費電力は0.439W(439mW)ということがわかった。

**8.回路設計**

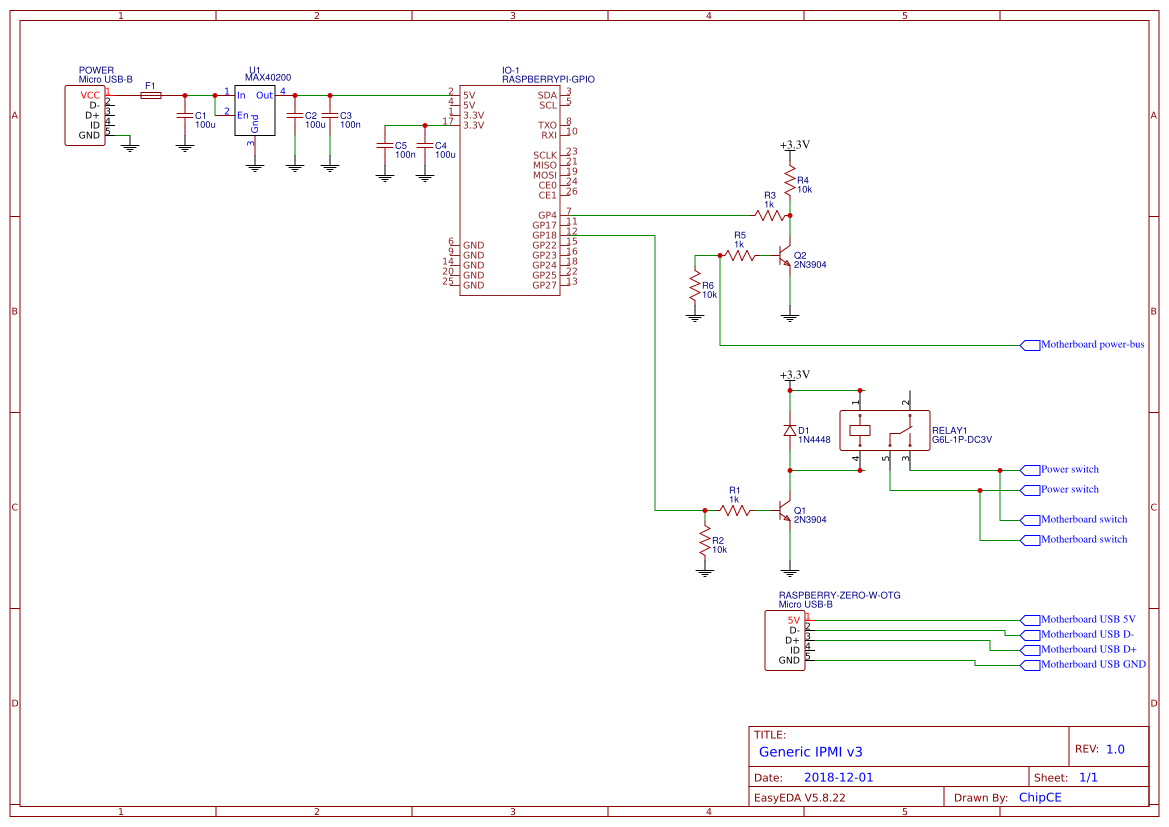


図15.IPMIモジュールの回路図

図15はIPMIモジュールの回路図を示す。

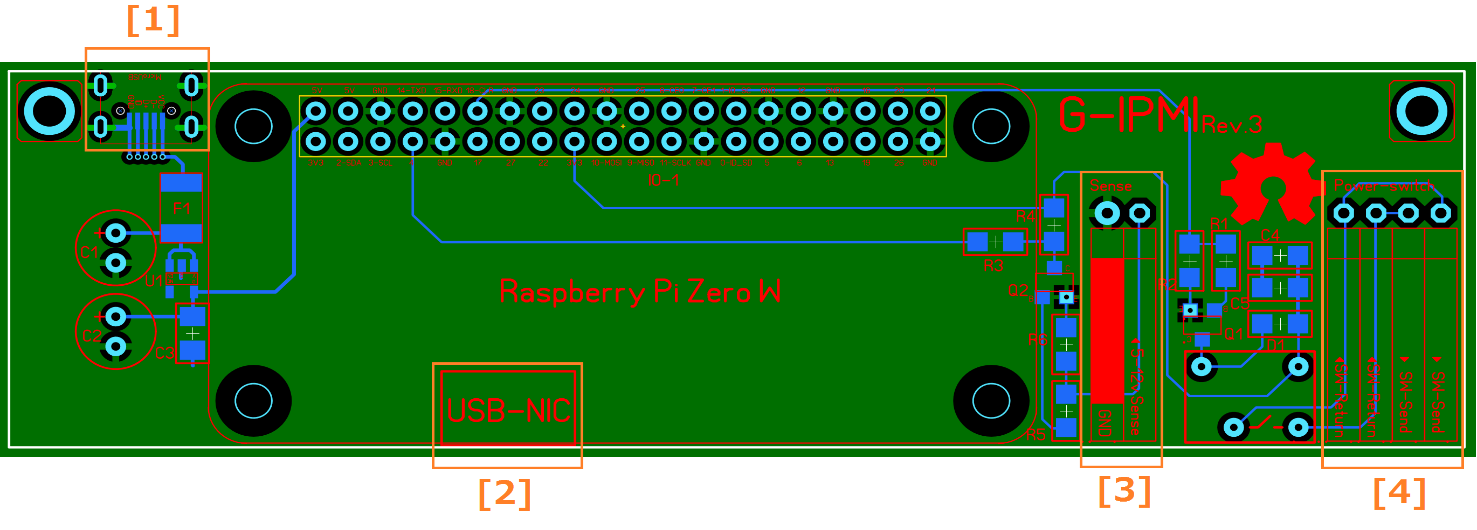


図16.基板配線図

　図16は基板配線図を示す。各コネクターを説明する。

* [1]電源入力:IPMIモジュールの電源を電源を供給する。
* [2]Raspberry Pi Zero WのUSBネットワークインターフェース:マザーボードのUSBヘッダに接続。
* [3]電源スイッチ制御コネクター:マザーボードの電源スイッチヘッダとケースの電源スイッチに接続。
* [4]電源状態監視コネクター:マザーボードの5V、または12Vバスに接続。

**9.まとめ**

本研究でRaspberry Pi Zero Wを用いてパソコン遠隔操作装置（IPMI）の構築を行った。作成した装置で、ホストPCの電源やネットワーク状態と関係なく遠隔操作を行うことができた。

　本研究で使用したRaspberry Pi Zero WはネットワークインターフェースカードとしてホストPCのマザーボードに接続するため、OSが起動する前のBIOS段階では電源制御以外の遠隔操作機能は使用できない。今後の課題として、BIOSの設定画面にも遠隔操作を行うことができれば、利便性が向上すると考えられる。