RPZ-PowerMGR

ファームウェア仕様書

2023/12/14

Indoor Corgi

1　状態遷移図

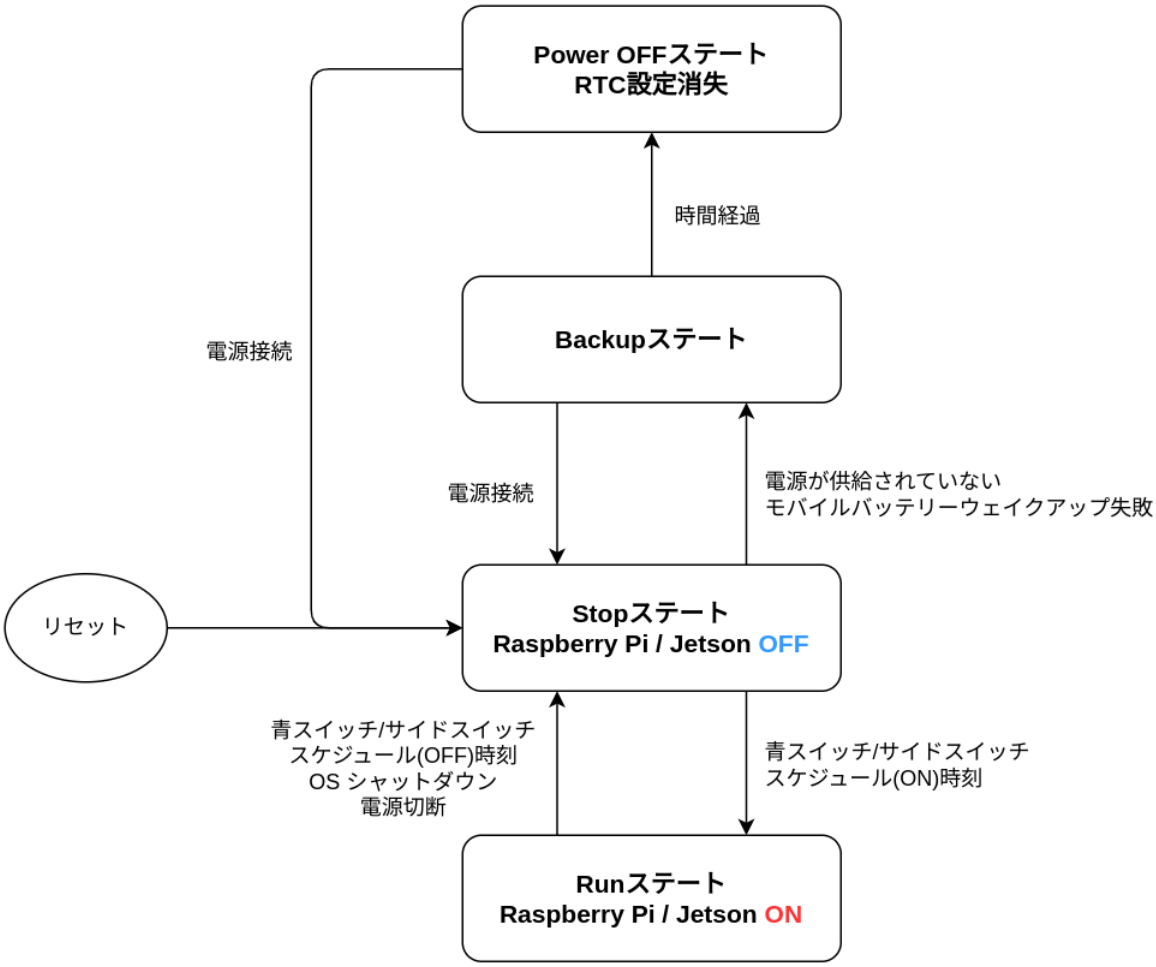

fig1: 状態遷移図

1.1　Stop ステート

初回電源投入後, もしくはリセット (SW1) 後は Stop ステートになります. Raspberry Pi / Jetson には通電しません.

1.2　Run ステート

Stop ステートで青スイッチ (SW2) かサイドスイッチ (SW3) が押される, ESW 外部入力を検出, またはスケジュールで設定した電源 ON 時刻が来ると, Run ステートに移行します. Raspberry Pi / Jetson は通電状態となります.

Run ステートで青スイッチ (SW2) かサイドスイッチ (SW3) が押される, ESW 外部入力を検出, スケジュールで設定した電源 OFF 時刻が来る, または Raspberry Pi / Jetson のシャットダウン完了を検知すると Stop ステートに移行します. 電源が切断された場合も Stop ステートに移行します.

1.3 Backup ステート

Stop ステートで電源が切断されるか, モバイルバッテリーのウェイクアップに失敗すると Backup ステートに移行します. Backup ステートではバックアップコンデンサーの電力を利用して時刻 (RTC) の保持動作のみを行います.

Backup ステートで電源に接続されると Stop ステートに移行します.

1.4 Power OFF ステート

Backup ステートのまま時間が経過し, バックアップコンデンサーの電力を使い切ると Power OFF ステートとなり, 全ての動作を停止します.

電源を接続すると Stop ステートに戻りますが, 時刻 (RTC) 情報が初期化されます. コンフィグ情報, 登録済みスケジュールは Power OFF ステートでも保持されます.

2 電源管理

2.1 電源 ON

電源 ON タイムチャートを fig2 に示します.

SW2/SW3/ESW 外部入力の立ち下がり, もしくはスケジューラーの電源 ON 要求がトリガーとなり, Raspberry Pi / Jetson を電源 ON に切り替えます. 入力電圧が低い場合はモバイルバッテリーのウェイクアップを試みます. Raspberry Pi / Jetson の電源と連動して LED2(緑) が点灯します.

電源 ON への切り替えと同時にスタートアップタイマーのカウントをスタートします. カウントが終了するまではスイッチやスケジューラーのシャットダウン要求を受け付けなくなり, Raspberry Pi / Jetson が起動するまでの時間を確保します. スタートアップタイマーのカウント中は LED1(RGB) が点灯します. 通常起動時は緑, ウェイクアップ信号で起動した場合は水色になります. スタートアップタイマーの時間は I2C で変更できます.

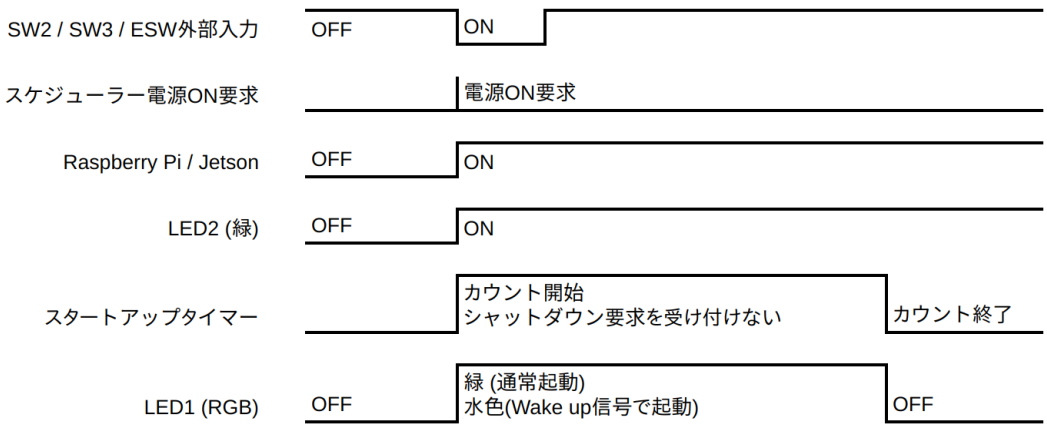

fig2: 電源 ON タイムチャート

2.2　電源 OFF(シャットダウン信号)

シャットダウン信号で電源 OFF するタイムチャートを fig3 に示します.

SW2/SW3/ESW 外部入力の立ち下がり, もしくはスケジューラーの電源 OFF 用要求がトリガーとなり, シャットダウン要求信号が HIGH/LOW トグルを繰り返します. Raspberry Pi / Jetson は自動的にシャットダウンを開始します. シャットダウン要求信号を使わず, ユーザー操作やプログラムによるシャットダウンでも構いません.

Version1.1 まではシャットダウン要求信号が LOW に変化するだけでしたが, エッジの検出漏れを防ぐため, Version1.2 からトグル (リトライ) するように変更しました.

シャットダウンが完了すると, Raspberry Pi からのシャットダウン完了信号が HIGH となります. Jetson からのシャットダウン完了信号は Hi-Z となり, RPZ-PowerMGR のプルアップにより HIGH となります. これを検知した RPZ-PowerMGR は, 再起動チェック時間内に再起動しないことを確認後, Raspberry Pi / Jetson の電源を OFF します. LED1(RGB) はシャットダウン要求信号を送信中黄色に点灯します.

シャットダウン要求/完了による電源 OFF の他, タイマーによる電源 OFF シーケンスも並行して処理されます. シャットダウン要求/完了信号は I2C で無効化できる他, GPIO16, 17, 26, 27 のいずれかに割り当ての変更が可能です.

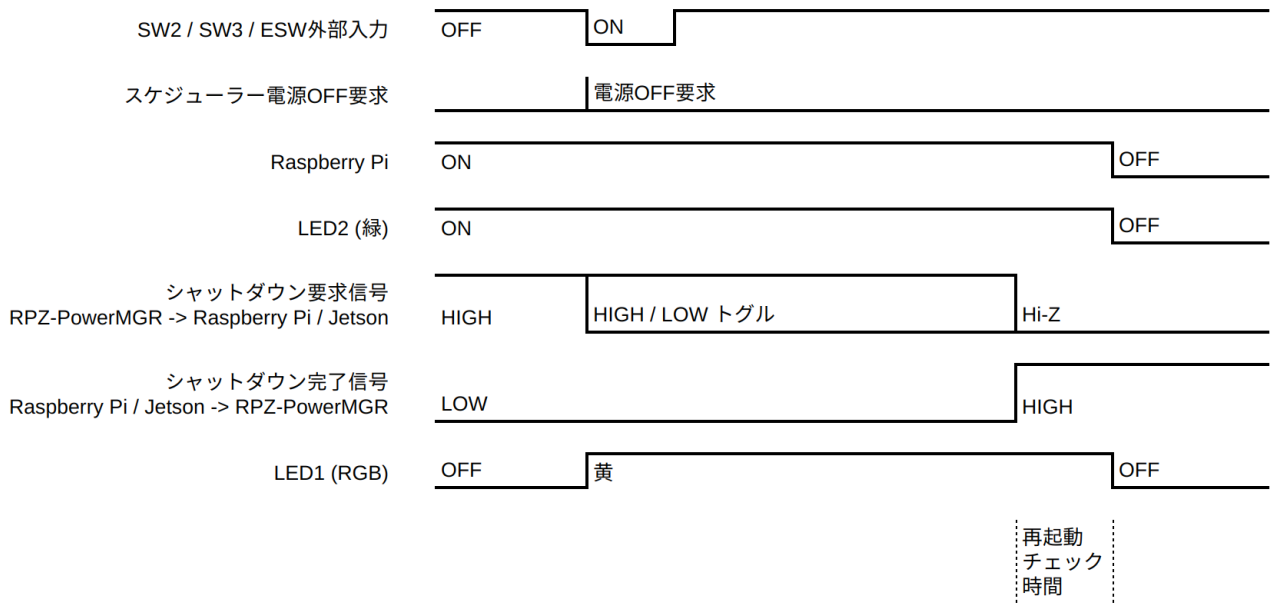

fig3: 電源 OFF タイムチャート (シャットダウン信号)

2.3　電源 OFF(タイマー)

シャットダウンタイマーで電源 OFF するタイムチャートを fig4 に示します.

SW2/SW3/ESW 外部入力の立ち下がり, もしくはスケジューラーの電源 OFF 用要求がトリガーとなり, シャットダウンタイマーのカウントが開始します.

タイマーのカウントが終了すると, Raspberry Pi / Jetson の電源を OFF します. 安全にシャットダウンするには, シャットダウン要求信号やユーザーによるシャットダウン操作が必要です. 本機能は Raspberry Pi / Jetson がフリーズした場合など不測の事態でも確実に電源を切ることを目的としています.

LED1(RGB) は並行して処理されるシャットダウン要求信号を送信中であれば黄色に点灯します．シャットダウン要求信号が無効な場合や送信に失敗してタイマーのみが稼働中の場合は紫色に点灯します．

シャットダウンタイマーの時間は I2C で変更できます．

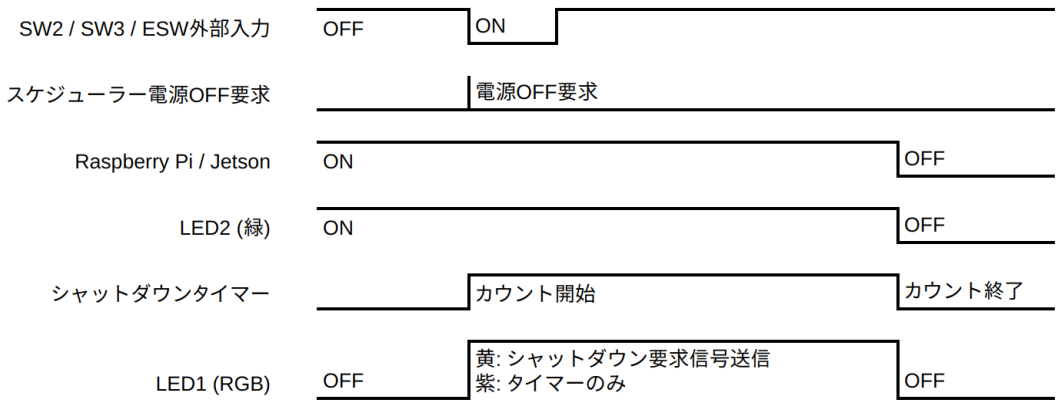

fig4: 電源 OFF タイムチャート (タイマー)

3 I2C

3.1 スレーブアドレス

プライマリー I2C アドレスは 0x20 です．DSW1-6 が ON の状態で Run ステートに移行した場合はセカンダリー I2C アドレス 0x22 に切り替わります．

RTC レジスターについては I2C アドレス 0x68 でも読み書きできます．これにより DS1307 互換デバイスとして動作可能です．

3.2 速度

Nomral mode 100kbps に対応しています．

3.3 レジスターアドレス

レジスターアドレスマップを fig5 に示します．2 バイトのデータ読み書きする際は下位アドレスから順に連続してアクセスして下さい．詳細は各項目を参照して下さい．

3.4 書き込み, 読み出し

I2C 書き込みフローを fig6 に, 読み出しフローを fig7 に示します．

Register Address	EEPROM Address	SRAM	Size (Byte)	Read/ Write	備考	
0x0	N/A	TimeManager	1	R/W	RTC 秒	RTC
0x1	N/A	TimeManager	1	R/W	RTC 分	コンフィグ
0x2	N/A	TimeManager	1	R/W	RTC 時	電源測定
0x3	N/A	TimeManager	1	R/W	RTC 曜日	スケジュール
0x4	N/A	TimeManager	1	R/W	RTC 日	コマンド
0x5	N/A	TimeManager	1	R/W	RTC 月	
0x6	N/A	TimeManager	1	R/W	RTC 年	
0x10 - 13	0x10 - 13		4	R	デバイスID: 0x52474D50	
0x14	0x14		1	R	ファームウェアマイナーバージョン	
0x15	0x15		1	R	ファームウェアメジャーバージョン	
0x16	0x16		1	R/W	スタートアップタイマー [秒] 1 - 255	
0x17	0x17		1	R/W	シャットダウンタイマー [秒] 0: 無効, 1 - 255	
0x18	0x18		1	R/W	シャットダウン要求信号 0: 無効, 1: GPIO16, 2: GPIO17, 3: GPIO26, 4: GPIO27	
0x19	0x19		1	R/W	シャットダウン完了信号 0: 無効, 1: GPIO16, 2: GPIO17, 3: GPIO26, 4: GPIO27	
0x1A - 1B	0x1A - 1B		2	R/W	タイムゾーン: UTCからの差分を分で指定. 日本は540.	
0x1C			1	R/W	電源自動リカバリー 0: 無効, 1: 有効	Version1.4 / 2.1で追加
0x1D			1	R/W	USB Type-A Wake up 0: 無効, 1: 有効	Version1.4 / 2.1で追加
0x20 - 21	N/A	Analog	2	R	Raspberry Pi最新の電流測定値 [mA]	
0x22 - 23	N/A	Analog	2	R	ログに記録された測定値の数	
0x24 - 25	N/A	Analog	2	W	ログから読み出す記録の番号. 0xFFFFを書き込むとログをリセット.	
0x26 - 27	N/A	Analog	2	R	指定した記録を読み出す [mA]	
0x30	0x30	TimeManager	1	R	登録されたスケジュールの数.	
0x31	N/A	I2C	1	R/W	読み出すスケジュール番号.	
0x32	0x100 + 4(n-1)	TimeManager	1	R/W	スケジュール 秒, 設定	
0x33	0x101 + 4(n-1)	TimeManager	1	R/W	スケジュール 分	
0x34	0x102 + 4(n-1)	TimeManager	1	R/W	スケジュール 日	
0x35	0x103 + 4(n-1)	TimeManager	1	R/W	スケジュール 月	
0x36	N/A	N/A	1	W	削除するスケジュール番号. 0xFFを指定すると全て削除.	
0x40	N/A	N/A	1	W	0xFFを書き込むとすぐにシャットダウン要求を実行. (スイッチを押した場合と同じ動作)	Version1.6 / 2.3で追加

fig5: レジスターアドレスマップ

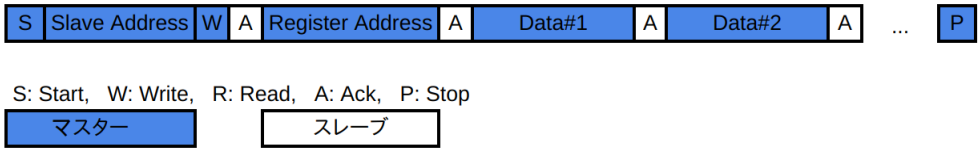

fig6: I2C 書き込み

4 RTC

4.1 RTC レジスター

RTC レジスターを fig8 に示します. 時刻は BCD フォーマットで指定します. DS1307 互換デバイスとして動作可能です.

4.2 スケジュール優先機能

RTC 時刻は手動設定以外にも, Raspberry Pi OS の NTP 同期により自動的に更新されます. そのため, OFF スケジュール時刻より先の RTC 時刻に更新された場合, OFF スケジュールが実行されないおそれがあります.

ファームウェア Rev1.6 / 2.3 以降において, 以下の条件を全て満たす場合には RTC 時刻が書き込まれても基板側の RTC 時刻は更新されません. NTP で時刻の誤差を修正する際に, 意図せずスケジュールがスキップされてしまうことを回避します.

RTC 現在時刻 < 次の OFF スケジュール時刻 < RTC 書き込み時刻

RTC 書き込み時刻 − RTC 現在時刻 < 10 分

上記式は, 後の時刻ほど大きい値という意味で記述しています.

5 スケジューラー

5.1 フォーマット

スケジュールのフォーマットを fig9 に示します. 時刻は BIN フォーマットで指定します.

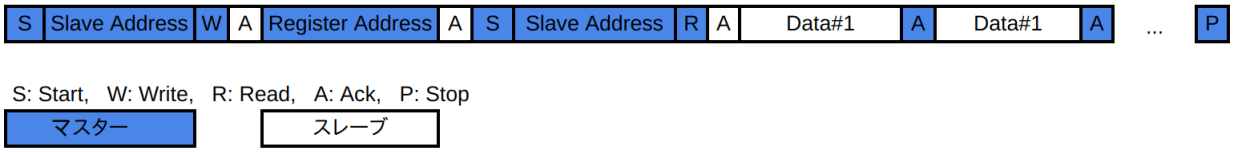

fig7: I2C 読み出し

Register Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	備考
0x00	0	秒 10の桁			秒 1の桁				
0x01	0	分 10の桁			分 1の桁				
0x02	0	0: 24h 1: 12h	0: AM 1: PM	時 10の桁	時 1の桁				
			時 10の桁		時 1の桁				
0x03						曜日			1: 日曜 ~ 7:土曜
0x04	0		日 10の桁		日 1の桁				
0x05	0			月 10の桁	月 1の桁				
0x06	年 10の桁				年 1の桁				西暦の下2桁

fig8: RTC レジスター

5.2 電源 ON, OFF

0x32 bit6 で電源を ON するか, OFF するかを指定します.

5.3 繰り返し

0x32 bit7 で Repeat(繰り返し) か One time(1 回のみ) かを指定します. One time を指定した場合, その指定した時刻に電源 ON/OFF が実行されると自動的に削除されます. ただし, 実行されなかった場合 (電源 ON スケジュール時刻に Run ステートだった場合など) は削除されません.

5.4 ワイルドカード

時, 日, 月については bit7 を 1 にすることでワイルドカードを指定できます. ワイルドカードを指定すると全ての値にヒットします. 例えば, 時をワイルドカード, 分を 30 とすると毎時間 30 分にスケジュールを実行します.

ファームウェア Version1.2 以前については, ワイルドカード指定に制限があります. 時をワイルドカードにした場合は日と月もワイルドカードにする必要があります. 日をワイルドカードにした場合は月もワイルドカードにする必要があります.

ファームウェア Version1.3 以降は上記の制限はありません.

5.5 スケジュール登録

スケジュールを登録するには, 0x32-35 にフォーマットに従ったデータを書き込みます. 既に登録されたスケジュールがある場合は末尾に追加されます. 最大で 250 個まで追加可能です.

<table>
<tr><th>Register Address</th><th>bit7</th><th>bit6</th><th>bit5</th><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th><th>備考</th></tr>
<tr><td>0x32</td><td>0 : Repeat
1: One time</td><td>0 : ON
1 : OFF</td><td colspan="6">Minutes, 0 - 59</td><td>bit7 : 継続か1回か指定.
bit6 : ON/OFF指定.
bit0 - 5 : 分を指定.</td></tr>
<tr><td>0x33</td><td>0 : Normal
1 : Wildcard</td><td colspan="2">0</td><td colspan="5">Hours, 0 - 23</td><td>時を指定. bit7=1でワイルドカード</td></tr>
<tr><td rowspan="2">0x34</td><td rowspan="2">0 : Normal
1 : Wildcard</td><td>0</td><td>0</td><td colspan="5">Date, 1 - 31</td><td>日を日付で指定. bit7=1でワイルドカード</td></tr>
<tr><td>1</td><td colspan="3">0</td><td colspan="3">Dow, 1 - 7</td><td>日を曜日で指定. 1が日曜. 7が土曜</td></tr>
<tr><td>0x35</td><td>0 : Normal
1 : Wildcard</td><td colspan="4">0</td><td colspan="4">Month, 1 - 12</td><td>月を指定. bit7=1でワイルドカード</td></tr>
</table>

fig9: スケジュールフォーマット

スケジュールを登録, もしくは RTC 時刻を変更すると, 登録済みスケジュールから最も近い時刻を再計算します. その際, スケジュール時刻まで 10 秒以下の場合は安定動作のためにスキップされる場合があります.

5.6 スケジュール読み出し

登録済みのスケジュールを読み出すには, 0x30 を読み出して登録済みのスケジュール数を確認します.

次に 0x31 に読み出すスケジュール番号を書き込みます. 番号は 1 から数えます. 最大値は 0x30 で読み出した数値となります.

0x32-35 を読み出すことでスケジュールの設定, 時刻を確認します.

5.7 RTC 未設定の場合

意図しない ON/OFF を避けるため, スケジューラーは RTC に時刻が設定されていない場合には動作しません. RTC が 2020 年以降に設定されている場合に設定済みと判断します. バックアップコンデンサーの残量が無くなった場合, RTC は 2000 年 1 月 1 日にリセットされ, スケジューラーは動作しません. 再度時刻を設定して下さい.

6 電流測定

6.1 概要

RPZ-PowerMGR には Raspberry Pi / Jetson の消費電流を測定する機能が搭載されています.

Raspberry Pi / Jetson を ON した時刻から 1 秒ごとに測定しいます. このうち, 最新の測定結果 1 個, および最初から 3600 秒分のログデータを保持しており, I2C で読み出すことができます.

6.2 最新の結果

アドレス 0x20-21 を読み出すことで最新の測定結果を [mA] を得られます.

6.3 ログデータ

アドレス 0x22-23 を読み出すことで記録されたログデータの数を得られます．読み出す記録の番号を 0x24-25 に書き込むと, 0x26-27 を読み出すことでデータを取得できます．番号は 0 から数えて指定します. (最大: ログデータの数-1)

0x24-25 に 0xFFFF を書き込むとログデータがクリアされて再度記録を開始できます.

Run ステートに移行時 (Raspberry Pi / Jetson 起動時) にログデータは毎回クリアされ, 記録が開始されます.

7 電源自動リカバリー

7.1 概要

通常の設定だと, バッテリーの電力が無くなったり停電によって電源が落ちてしまった場合, 手動でスイッチを押すか, 次の電源 ON スケジュール時刻を待つ以外に Raspberry Pi / Jetson を起動する手段がありません.

また, 電源が落ちた状態が数日以上続くと, バックアップコンデンサーの電力がなくなり, RTC の時刻が初期化されます．この状況では, 手動でスイッチを押す以外に Raspberry Pi / Jetson を起動することができません.

そこで自動的に Raspberry Pi / Jetson の電源を復帰する「電源自動リカバリー」機能を搭載しました. これにより, 意図しない電源切断やバッテリー切れとなっても, スイッチ操作なしで再度 Raspberry Pi / Jetson を起動し, 運用を再開することが可能となります。

7.2 設定方法

デフォルトでは無効になっています．レジスターアドレス 0x1C に 1 を書き込むことで有効化できます.

7.3 状態遷移図

fig10 の赤枠で囲われている箇所の動作が変更されます．無効時は Power OFF および Backup ステートから電源接続されると Stop ステートに移行します．有効時は電源リカバリー条件を満たしている場合, Run ステートに移行 (Raspberry Pi / Jetson 電源 ON) します.

8 USB Type-A Wake Up

8.1 概要

モバイルバッテリーの多くは, 一定時間使用しないとスリープ状態に入り, 電圧を出力しない状態になります．そこで, RPZ-PowerMGR は, 電圧が低いと状況に応じてモバイルバッテリー Wake up を試みます．これはモバイルバッテリー以外が電源に接続されていても発生します．(何が繋がっているか RPZ-PowerMGR 側からは判別できないため)

ところが, RPZ-PowerMGR と電源がケーブルで繋がっているものの電気が来ていない状態だと, Wake up の副作用で RTC がリセットされてしまう現象が発生することがあります．この現象が発生するかどうかは電源の回路に依存します.

そこで, 上記現象を回避するために USB Type-A 型のモバイルバッテリー用の Wake up を無効化する機能を搭載しました．USB Type-C 型の Wake up はこの問題が発生しないため, 設定に関わらず常時有効です.

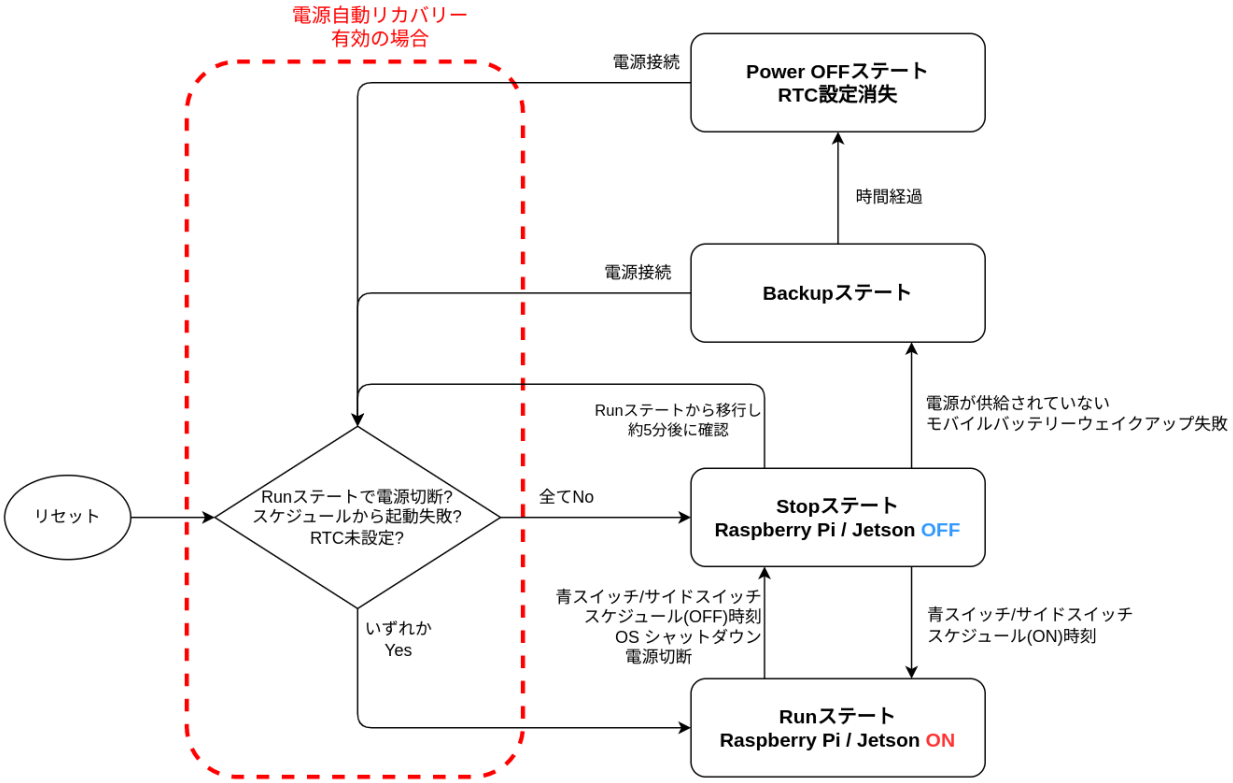

fig10: 電源自動リカバリー有効時の状態遷移図

8.2 設定方法

デフォルトでは有効になっています. レジスターアドレス 0x1D に 0 を書き込むことで無効化できます. 無効化されるのは USB Type-A 型のモバイルバッテリー向けの Wake up 機能です.

9 ファームウェアバージョン

RPZ-PowerMGR 初期版の Rev1 と改良版の Rev2 の 2 種類のハードウェアがあるため, ファームウェアを分けています.

ファームウェア Version1.x がハードウェア Rev1 用, ファームウェア Version2.x がハードウェア Rev2 用となります.

Rev1 ではファームウェア Version1.5 以降, Rev2 ではファームウェア Version2.2 以降で Jetson Nano に対応しました.

10 更新履歴

2021/1/6　初版リリース.

2021/2/22　fig5 でファームウェアメジャーバージョンとマイナーバージョンの アドレスが逆になっていた誤記を修正.

2021/4/1　Version1.2 から fig3 のシャットダウン要求信号をトグル (リトライ) するように変更.

2021/5/24　Version1.3 からワイルドカード指定制限がなくなりました. 月, 日, 時に自由な組み合わせでワイルドカードを指定できます.

2021/11/5　Rev2 で追加されたサイドスイッチ SW3 を追記.

2022/1/31　ファームウェア本体のバージョンと混同しないよう, 仕様書のバージョンを削除し, 日付のみに変更. スケジューラーの動作に RTC の設定が必要であることを明記.

2022/7/3　ファームウェアバージョン 1.x と 2.x の違いを明記. 新しく追加された電源自動リカバリー, USB Type-A Wake up の説明とアドレスを追記.

2023/6/6　Jetson Nano 対応したファームウェアバージョン 1.5, 2.2 に対応.

2023/7/4　ファームウェアバージョン 1.6, 2.3 で追加された, I2C 通信によるシャットダウン要求, RTC 時刻書き込み時のスケジュール優先機能を追記.

2023/12/14　電源 OFF(シャットダウン信号) において, 再起動チェック時間をタイムチャートに追記.