

給電の中断を防ぐための スマートなバッテリー・バックアップ 【Part2】 BBUのマイクロコントローラ の機能と動作

著者：Christian Cruz、スタッフ・アプリケーション開発エンジニア
Marvin Neil Cabueñas、シニア・ファームウェア・エンジニア

概要

Open Compute Project (OCP) は、データ・センター向け製品の設計とベスト・プラクティスを企業間で共有できるようにすることを目指す非営利組織です。Open Rack Version 3 (ORV3) は、そのOCPが公開した仕様です。このバージョンでは、最も重要な変更点として12Vのアーキテクチャから48Vのアーキテクチャへの移行が行われました。本連載では、5回にわたりアナログ・デバイスが開発したバッテリー・バックアップ・ユニット (BBU: Battery Backup Unit) のリファレンス設計について解説しています。前回の記事「給電の中断を防ぐためのスマートなバッテリー・バックアップ【Part1】電気設計と機械設計」では、BBUの電気設計と機械設計について考察しました。今回 (Part2) は、メインのマイクロコントローラに実装するソフトウェアについて詳しく解説することにします。そのソフトウェアは、各種プロセスの円滑な動作を保証し、BBUの高い効率と容量を確保するための役割を果たします。定められた仕様を満たすことが可能なシステム・レベルのソリューションを実現するには、ハードウェアとソフトウェアを円滑に協調させる必要があります。

はじめに

BBUモジュールでは、ハウスキューピングに必要な処理が適切に行われるようにする必要があります。それにより、無視できない多くのメリットが得られます。整然とした回路を採用すれば、より高い安全性がもたらされるという大きなメリットが得られます。それにより、電気的な原因でBBUモジュールに火災などの危険が及びリスクが低下します。また、適切なハウスキューピングは、正常な動作を妨げるおそれのある電気的な障害を防ぐことにつながります。このことは、回路の性能の向上と寿命の延伸に寄与します。回路の監視と調整を定期的に行い、起こり得る問題や誤動作が見つかったら、その原因を特定して適切に対処することが非常に重要です。回路の適切なハウスキューピングを優先することは、高い安全性と優れた性能を確保するためのシンプルながらも効果的な手段になります。このことは、決して見逃されるべきではありません。

各BBUモジュールには、メインのマイクロコントローラと、バッテリー管理 (バッテリー・マネージメント) システム (BMS: Battery Management System) 用のマイクロコントローラが実装されます。アナログ・デバイスのリファレンス設計では、Arm® ベースで超低消費電力のマイクロコントローラ「MAX32690」をメインのマイクロコントローラとして使用しています。

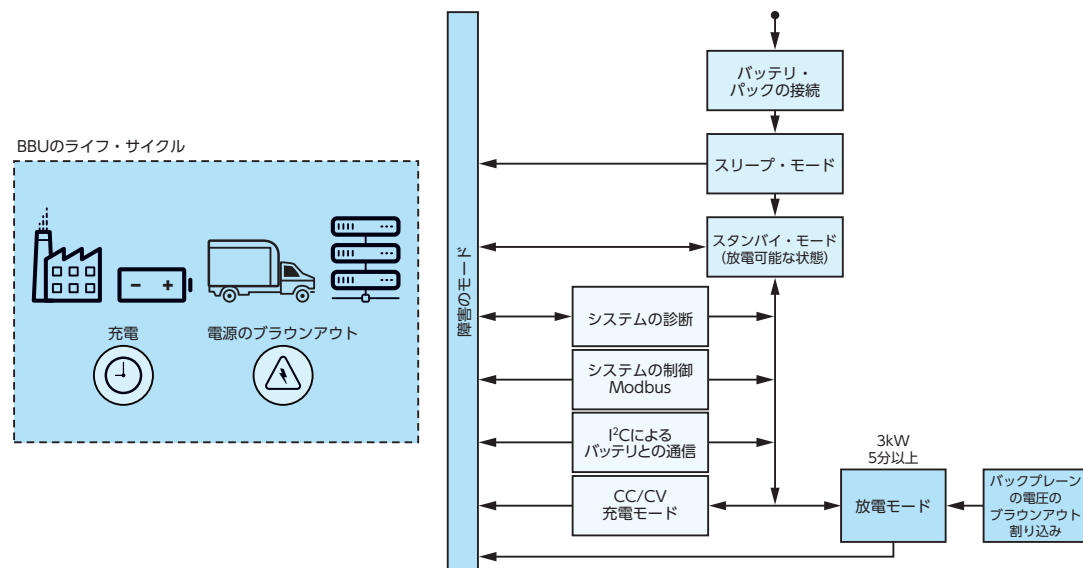


図1. メインのマイクロコントローラによる
ハウスキーピングの動作サイクル¹⁾

このマイクロコントローラは、6つの重要なプロセスを管理します（図1）。6つのプロセスとは、以下のようなものです。

- ▶ I²C を介したハウスキーピングと様々なペリフェラルに対する通信
- ▶ バックプレーンの電圧に関するデータに基づいた放電シーケンスの処理
- ▶ 定電流モード／定電圧モードによるバッテリーの充電処理
- ▶ BBU モジュールにおける充電状態と放電状態間の遷移
- ▶ 障害の処理と対応
- ▶ Modbus のコマンドに対する Modbus のフォローとしての対応

また、このリファレンス設計では、BMS用のマイクロコントローラとして「[MAX32625](#)」を使用します。このマイクロコントローラは、マルチセル・バッテリー・モニタ「[ADBMS6948](#)」との通信と、すべての測定データをメインのマイクロコントローラに送信する役割を担います。

メインのマイクロコントローラが担う処理

ここからは、メインのマイクロコントローラが担う処理について詳しく説明していきます。

I²Cを介したハウスキーピングと様々なペリフェラルに対する通信

BBUモジュールの回路には、数多くの補助デバイスが接続されています。メインのマイクロコントローラは、それらに対してI²Cのメインとして振る舞います。同マイクロコントローラは、補助デバイスからのデータを収集して保存します。また、様々なタスクを実行するために、I²Cに対応する様々な補助デバイスを制御する役割も担います。補助デバイスとしては、具体的には以下のようなものがあります。

- ▶ [MAX32625](#) : BMS 用のマイクロコントローラ
- ▶ [LTC2971](#) : 2 チャンネルのパワー・システム・マネージャ
- ▶ [MAX31760](#) : 高精度のファンスピード・コントローラ
- ▶ [24AA512T \(EEPROM\)](#) : 重要なデータの取得／保存用のデータ・ストレージとして機能
- ▶ [LTC2991](#) : 温度センサー

BMS用のマイクロコントローラ

メインのマイクロコントローラは、BMS用のマイクロコントローラ (MAX32625) との間で定期的に通信を実行します。それにより、個々のバッテリー・パックのセル電圧、SOC (State of Charge)、SOH (State of Health)、セルの温度に関する最新の情報を取得します。それだけでなく、バッテリー・パックで発生した可能性がある障害に関する情報も取得します。セルの電圧、SOC、SOH、温度は、急激に変化することはないはずです。そこで、これらの情報は4分ごとに更新するように実装されています。何らかの障害が発生した場合には、2つのマイクロコントローラの共有ピンがハイになります。そして、メインのマイクロコントローラに対しては、BMS用のマイクロコントローラの障害情報を直ちに読み取ることを求める割り込みがトリガされます。2つのマイクロコントローラの間の迅速な通信を可能にするために、メインのマイクロコントローラには、BMS用のマイクロコントローラとの通信専用のI²Cポートが設けてあります。

パワー・マネージメントIC

メイン・マイクロコントローラは、PMBus (Power Management Bus) のプロトコルを介してパワー・マネージメントICであるLTC2971と定期的に通信を行います。それにより、電圧、電流、温度の測定値をやり取りすると共に、警告、障害の確認を実施します。メインのマイクロコントローラは、ルーチンを調整するためにバックプレーンの出力電圧の値をパラメータとして測定します。その際、LTC2971はクイック・フィードバック用の機能を提供します。また、放電動作モードでは、出力電圧が必ずレギュレーションの範囲内に収まるように調整を行います。パワー・コンバータのフィードバック電圧を調整すれば、出力電圧を1%引き下げることができます。

ファンスピード・コントローラ

MAX31760は、ファンスピード・コントローラです。このICは、BBUモジュールのファンの速度を調整する役割を果たします。マイクロコントローラによってPWM (パルス幅変調) のデューティ・サイクルが設定され、I²Cを介してファンの速度が調整されることになります。メインのマイクロコントローラは、バックプレーンの負荷電流とバッテリー・パックの負荷電流のいずれか一方の温度に基づいて、必要なファンの速度を計算して調整を行います。

データ・ストレージ

オンボードのEEPROMは、BBUモジュール全体の外部ストレージ・デバイスとして機能します。バッテリーの電圧レベル、SOC、SOH、セルの種類／モデル年式、ボードの温度といった重要な情報は、メインのマイクロコントローラがEEPROMのフラッシュ・ページにI²Cを介して書き込みを行うことによって定期的に保存されます。ユーザは、1時間ごとに更新されるこれらのデータを参照することで、メンテナンスやトラブルシューティングを効果的に実施できます。

オンボードのデジタル温度センサー

LTC2991は、8個の測定回路を備えるセンサーICです。電圧、電流、温度を監視する用途に使用できます。バッテリー・モジュールの温度の測定は、その内部に戦略的に配置されたデジタル・センサーによって行われます。本稿で紹介しているリファレンス設計では、そのセンサーとしてLTC2991を使用しています。それによって測定された温度の値に基づき、メインのマイクロコントローラは、パワー・ボードとバッテリー・スタックの動作温度を維持します。具体的には、決して40℃に達することがないようにファンの速度を調整します。

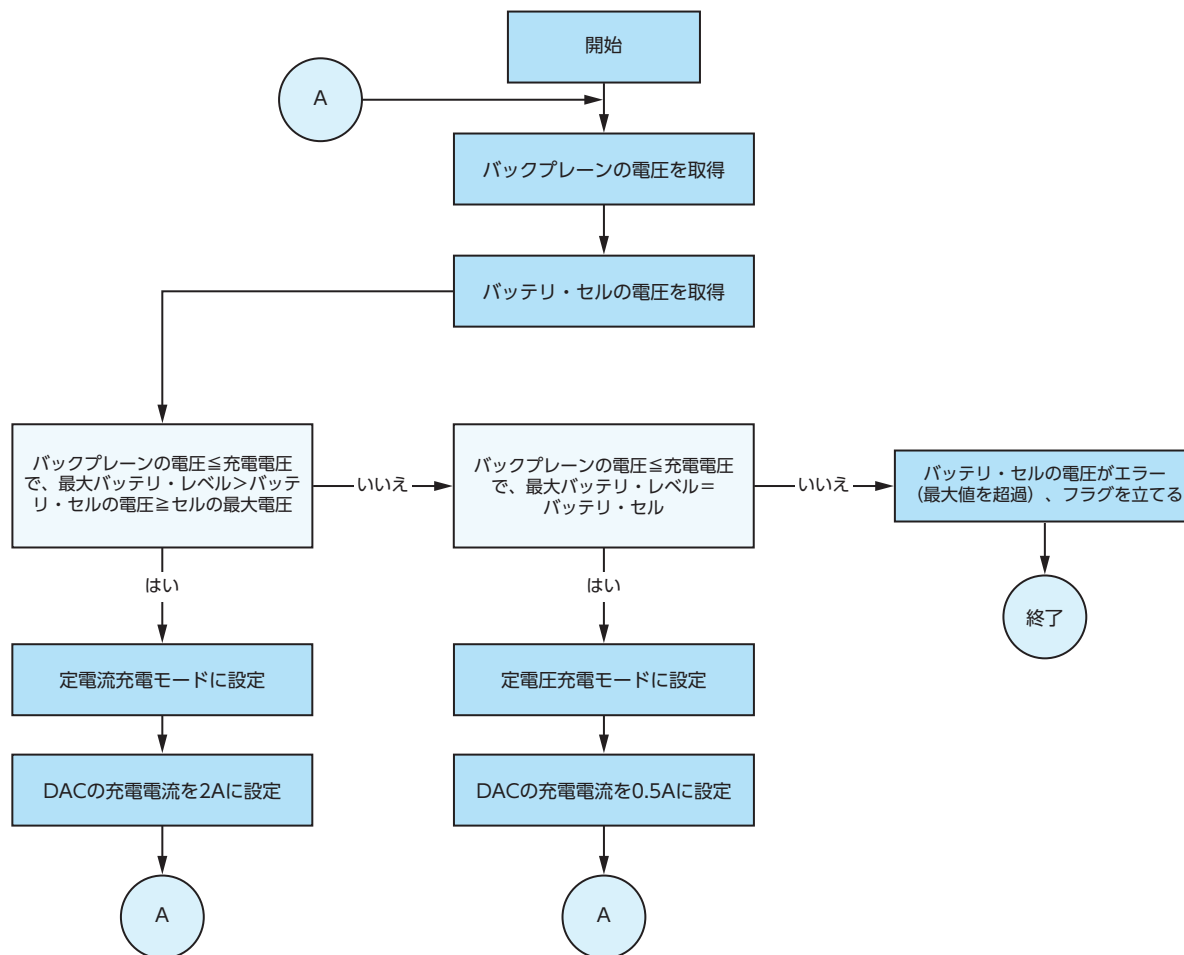


図2. 定電流／定電圧の充電アルゴリズム

定電流／定電圧の充電方法を管理する

メインのマイクロコントローラは、バッテリーの充電モードを管理する役割も担います（図2）。初期状態において、バッテリー・セルの残容量が非常に少ない場合、マイクロコントローラは5Aの最大電流でバッテリーを充電することを許可します。セルが定常状態になったら、マイクロコントローラは充電電流を調整します。具体的には、個々のセルの電圧が4Vに達していなければ充電電流を2Aに設定します。また、BMS用のマイクロコントローラから送られてくるデータに応じ、バッテリー・セルの電圧を定期的にモニタリングし、ペリフェラルのA/Dコンバータを使用してバッテリー・プレーン全体の電圧を測定します。そのようにすることで、メインのマイクロコントローラは、すべてのセルが4Vに達したかどうかを確認します。そして、すべてのセルが4Vに達したら、充電モードを定電圧モードに切り替えます。その際には、充電電流をわずか0.5Aまでに制限します。つまり、メインのマイクロ

コントローラは、バッテリー・プレーン全体の電圧レベルを継続的にモニタリングしつつ、BMS用のマイクロコントローラからのデータを取得して、すべてのバッテリーが満充電になったかどうかを確認します。

充電時／放電時のパワー・コンバータの振る舞い

停電が発生した際、BBUはどのようにしてスタンバイ・モードからバッテリーへの給電モードに遷移するのでしょうか。この点について十分に理解しておくことが肝要です。BBUのメインのマイクロコントローラは、予期せぬ電力の喪失やデータの喪失を防ぐために、バックプレーンの電圧レベルを入念にモニタリングします。本稿のリファレンス設計では、100Vの入出力に対応する昇降圧コントローラ「LT8228」を採用しています。そのデフォルトの充電モードは、約49V～53Vに設定されています。

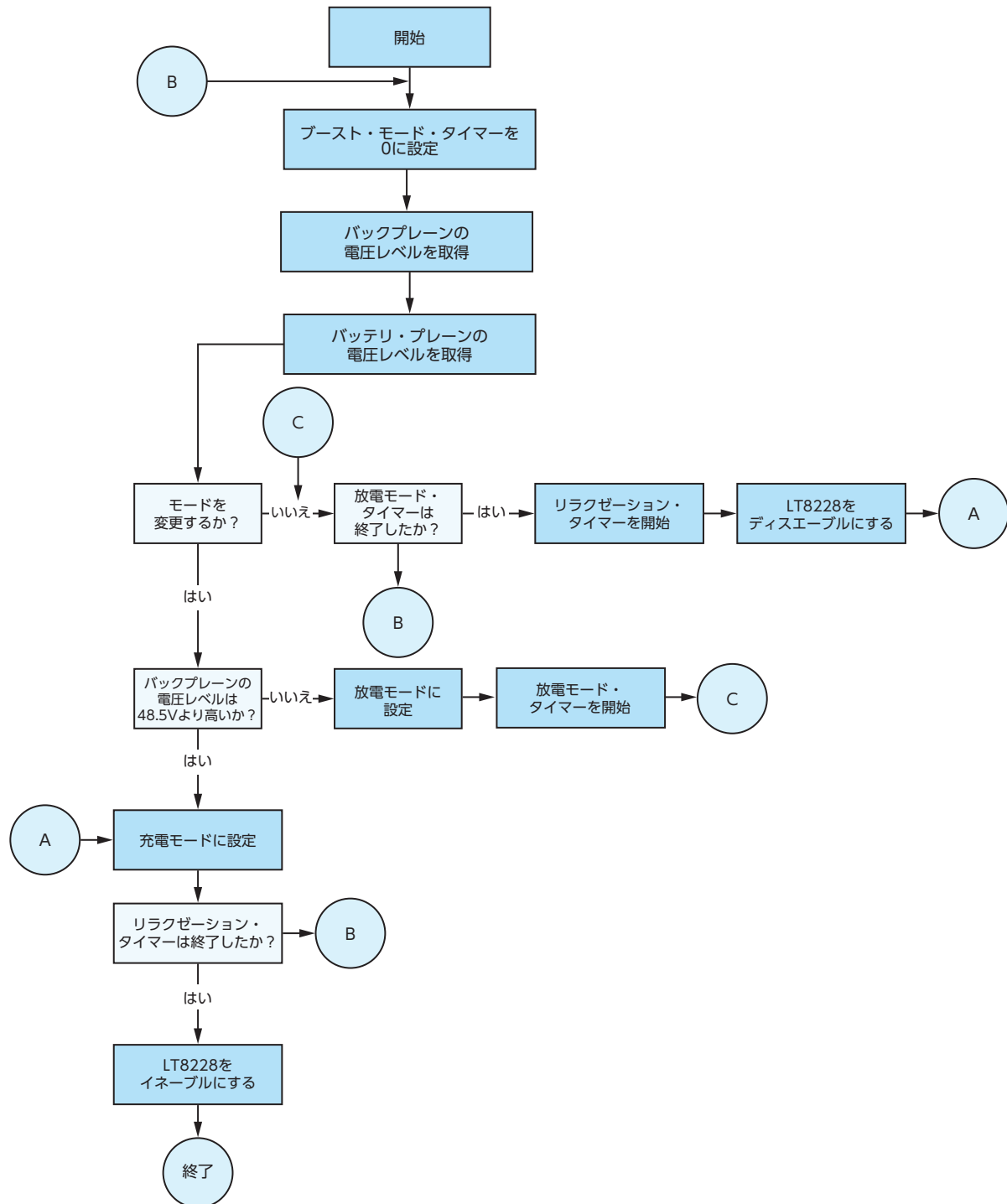


図3. BBUモジュールの充電／放電制御

メインのマイクロコントローラは、バックプレーン電圧が48.5Vを下回る状態が2ミリ秒続いたことを検出すると、直ちにLT8228の方向ピンを充電モードから放電モードに切り替え、その状態を4分間保つことで停電に対応します。BBUは、4分が経過するまで放電を続けることによって給電の中断を防ぐということです。4分が経過してもバックプレーン電圧が回復せず、バッテリー・セルがそれに耐えられる状態であれば、更に1分間待つことでセルをクールダウンし、放電モードへと移行します。電力が復旧したら、BBUは再びメインの電源に切り替えてバッテリーの再充電を開始します。図3は、BBUにおける充放電制御についてまとめたものです。

障害の処理と対応

障害については、どのように対応するのかという計画を明確に定義しておくことが不可欠です。潜在的な問題を事前に洗い出し、それらに対処するための手順を確立しておく必要があります。そうすれば、ストレスや苛立ちを感じるケースも少なくなるでしょう。また、対象とするシステムに、各種のデバイスと通信する手段を組み込んでおくことも重要です。それにより、何が起こり得るのかを把握している状態を保ちながら、システムを円滑に運用することが可能になります。誤りや障害というのは、実際に起こり得るものです。それらにどのように備えておくのかということが、最終的な結果に大きな影響を及ぼします。

OCPは起こり得る障害の種類も示しています。そのことも踏まえつつ、リファレンス設計で使用するメインのマイクロコントローラは、障害に対する誤警報の発生確率を低減できるように構成されています。そのファームウェアは、障害の引き金となる異常を検出するために、アルゴリズムによるチェックを行うよう実装されています。異常が検出された場合には、検証の処理が実行されます。その問題が連続して発生する場合や、設定済みのサイクル数の間に問題が発生する場合には、その異常は障害であると判断されます。これは、本当の障害だけを特定し、それを解消することを保証するための賢明な方法だと言えます。

Modbusのコマンド、フォローとしての動作

Modbusの機能コードは、Modbusに対応するネットワーク内のデバイス間で通信を確立するための必須の要素です。このコードによって、デバイス間でやり取りされるリクエスト／レスポンスの種類（データの読み出しや書き込みなど）が決まります。よく使用されるコードとしては、モジュール・メトリクス、モジュール動作ステータス、モジュール・レジスタ・アドレス、障害ステータスなどが挙げられます。

Modbusのコマンドの処理は、メインのマイクロコントローラの制御下で行われます。メインのマイクロコントローラは、OCPの仕様で必要だと見なされているModbusの基本的なコマンドのすべてをサポートします。それらのコマンドにより、モジュールの設定を変更したり、バッテリーのSOC／SOH／セルの電圧レベル／充放電電流などのパラメータに関する重要な情報を取得したりすることができます。メインのマイクロコントローラは、メッセージを検証した上で、受信したコマンドに応じて動作します。

ここで、図4をご覧ください。ご覧のように、メインのマイクロコントローラは、トランシーバーIC「ADM2561」、「ADM3061」を利用できるようになっています。それにより、UART (Universal Asynchronous Receiver Transmitter) のプロトコルを介して、様々なペリフェラルからのデータを処理し、BBUシェルフのマイクロコントローラに送信します。独立したトランシーバーを利用することには、システム・レベルの電磁干渉の影響を回避し、OCPが定める電磁環境適合性（EMC）の基準を順守することが可能になるというメリットがあります。

また、BBUシェルフからホストPCへの通信には、ADM2561を使用します。同ICは、DB9（D-Subminiature）コネクタとRJ45のインターネット・ポートを介してホストPCに接続されています。

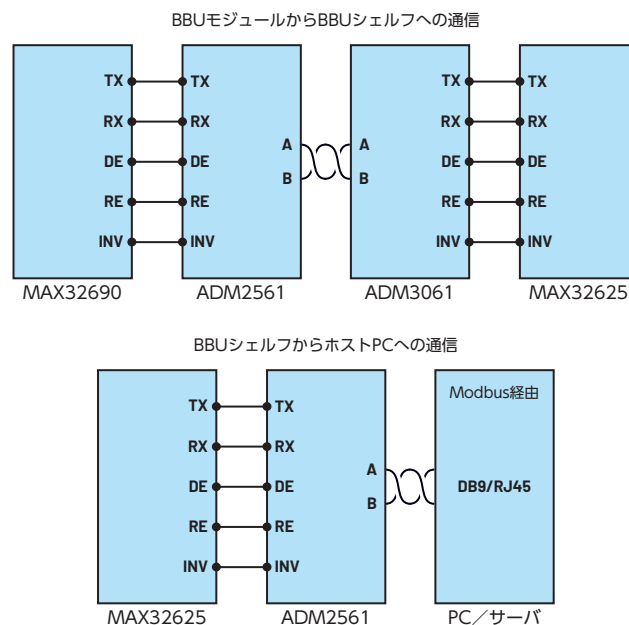


図4. モジュールからシェルフ、シェルフからPCへの通信に用いる構成

まとめ

コントローラのアーキテクチャには、堅牢性、生産性、卓越した適応性が求められます。これらは欠かせない要件だと言ってもよいでしょう。そのためには入力と出力を明確に定義する必要があります。また、高度なアルゴリズム・ルーチンを処理する能力を用意しなければなりません。加えて、アルゴリズムをシームレスに変更／更新できるようにするためには、アーキテクチャをモジュール式で構成する必要があります。更に、システムの障害や事故を防止しつつ予期せぬ事象に容易に対応できるように、安全機構を適用しなければなりません。コントローラのアーキテクチャの質を高めるには、誤りや誤動作の確率を抑えつつ、信頼できる効果的なシステム制御を提供できるようにする必要があります。

次回（本連載のPart3）は、BBUのBMSとそれに付随するマイクロコントローラについて解説する予定です。その中では、バッテリー・パックの個々のセルの電圧とバッテリー・パックの温度を維持するために、取得可能なデータをどのように処理／使用すればよいのかを明らかにします。また、セル・バランシングや、SOC/SOH用のアルゴリズムの実装、それらの情報を使用してバッテリー・パックのライフ・サイクルを延長する方法などについて解説します。

参考資料

¹ David Sun「[Open Rack/SpecsAndDesigns \(Open Rack - 仕様と設計\)](#)」Open Compute Project



著者について

Christian Cruzは、アナログ・デバイセズ（フィリピン）のスタッフ・アプリケーション開発エンジニアです。2020年に入社しました。現在は、クラウド・ベースのコンピューティングおよびシステム通信アプリケーションを対象としたパワー・マネージメントに関するサポートを担当。12年間にわたり、アナログ/デジタル設計、ファームウェア設計、パワー・エレクトロニクス設計（AC/DC、DC/DCの電力変換を含むパワー・マネージメントICの開発）に携わってきました。フィリピン マニラのザ・イースト大学で電子工学の学士号を取得しています。



著者について

Marvin Neil Solis Cabueñasは、アナログ・デバイセズのシニア・ファームウェア・エンジニアです。パワー事業部門で様々なプロジェクトを担当。組み込みシステムのプログラミング、デジタル信号処理、シミュレーション用のモデリングなど、様々な分野における9年以上の経験を有しています。2021年に入社する前は、Azeus Systems Philippinesにシステム・エンジニアとして勤務。2014～2017年にはTechnistock, Philippinesでネットワーク・エンジニア、2017～2020年にはNokia Technology Center Philippinesで研究開発エンジニアとして業務に携わっていました。フィリピン マニラのデ・ラ・サール大学で電子工学の学士号を取得。現在は、フィリピン大学で電気工学の修士号の取得を目指しています。