

スマートグリッド向け要素技術の調査研究（まとめ）

～消費電力の見える化技術～

西田 龍己* 若生 進一* 平間 毅* 戸塚 貴之*

1. はじめに

2011年3月11日の震災以降、3年が経過するも東京電力管内では電力供給不足が慢性化し、電力料金の高騰へとなっている。県内中小企業でも電力消費への関心は高く、電気の使い方を見直す活動は続けられており、そのためのツールとして電力見える化システム、特に各分電盤や各電気機器の消費電力を簡易的に把握できるシステムに対するニーズは高く、本システムと生産活動を連携させ更なる業務効率向上を目指す活動は以前にも増して高まっている。

このような背景から、当センターではH23年度、H24年度とセンシング技術、マイコン技術、無線通信技術等の基盤技術を活用して、各分電盤や電気機器の消費電力を簡易的に測定できるシステムを試作し、電力見える化システムの試作・実証実験を行ってきた。H25年度はその集大成として、企業での実証実験を継続しニーズを抽出すると共に、そのニーズを反映したアプリケーション開発を進め更なる利便性改善を図ったシステムへの改良を行った。

2. 目的

本調査研究では、今まで試用頂いたシステムを企業独自で導入可能な様に汎用化を図ると共に利便性を改善し、企業内で様々な用途で活用頂く事を目的としてアプリケーションの改良を図った。

3. 研究内容

今まで活用してきたシステムは、下記の様な処理構成となっており、シンプルではあるものの企業独自での設定においては一部の専門知識及び関連ツールも必要であった。

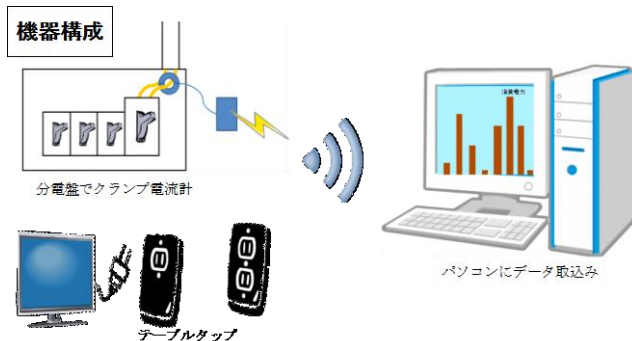


図1 電力見える化システムの概要

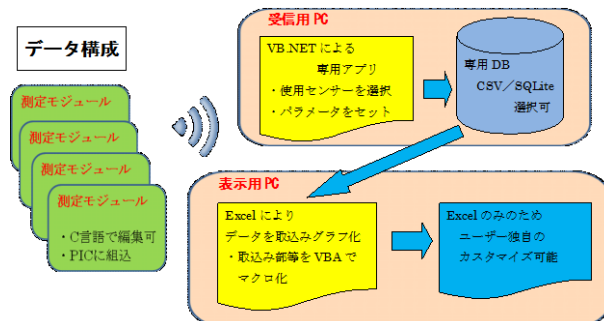


図2 データ処理の基本構成

そこで、機器構成（図1）及びデータ構成（図2）の基本構成は変更せず、一部のアプリケーションの改良のみで、企業での独自設定を可能としたシステムへの改善を図る事とした。

また、ハード、ソフト含めたユーザーマニュアルを作成し専門家のサポートを必要とせず有る程度の電気、パソコンの知識があればユーザー独自で操作・運用が出来る事を狙った。

3.1 モジュール内PICのプログラム標準化

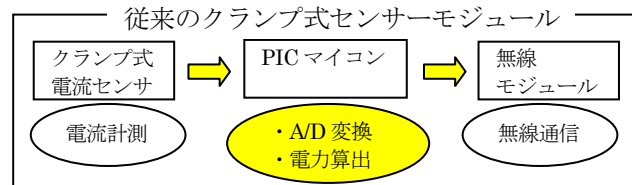


図3 センサーモジュールの処理

今まで活用してきたモジュールでの処理は、図3の如くPICマイコンにて、A/D変換後、電力算出までを行っていたが、これにより装着するセンサータイプ（図4）、電流範囲を決める並列抵抗毎に、設定値を変更する必要がある、その都度マイコンプログラムの書換えが必要で有った。そこでセンサーモジュール側では計測した電圧値（実行値）をそのまま出力し、受信側のプログラムにて電流値及び電力値に変換する事にし、マイコンのプログラム書換えを不要とした。

尚、電流計測以外の場合にはマイコンのプログラムの修正は必要となるが、その場合でも受信側では変換係数を設定出来るので、マイコンプログラムと連携し様々な用途で対応できる。



図4 各種電流センサ

3.2 受信プログラムの改善

前記のマイコンプログラム変更に伴い、受信側アプリケーションの変更も必要となるが、この変更にあわせてアプリケーション形態も改善する事とした。

従来は、図 2 に示す、“受信用 PC”を、Windows PC 以外も想定し、アプリケーションを『python』にて作成していたが、今回の実証実験で協力頂いた企業全てで、Windows PC が導入されていた事とその後でデータ処理（グラフ化）では Micro Soft 社の Excel が必要になる事もあり、利便性改善も考慮して Micro Soft 社の VB.NET を用いて開発する事にした。

これにより、アプリケーションもビジュアル化出来、操作性が改善出来ると共に、様々な追加メニューに対応でき利便性を大幅に改善する事ができた。

図 5 に変換パラメータ設定画面を示すが、ここでは各モジュールの端子毎に、装着センサ、パラメータ、電圧値等を設定し、受信アプリケーションプログラムにて電力換算しデータベースの保存する仕組みとなっている。

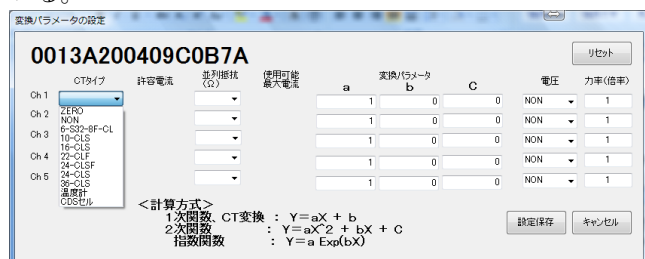


図 5 変換パラメータの設定画面

3.3 キュービクルでの電力収集

キュービクル内での電流計測は高電圧の為、電流センサ設置には有資格者による作業が必要で、ユーザー自身でのセンサ設置は困難である。

そこで、キュービクル内の電力計より電力量情報を出力するケーブル設置工事を電力会社にお願ひし、ユーザー側ではこのケーブルのパルス信号計測で電力の把握が可能となる。

このパルスをクランプ式で計測する装置は市販されているが、今回の試作モジュールでは、信号取込み端子部に 5V 電源部も設けており、これを用いて信号増幅する事で、一般の交流電流センサを用いたパルス計測を試みた。(図 6)

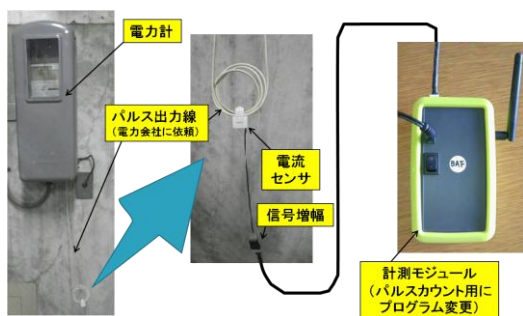


図 6 電力パルス検出例

ここで信号増幅部では、5V で動作可能なオペアンプを用いて電流センサの信号を増幅し、計測モジュール

ールではマイコンプログラムを変更し、信号の立ち上がり回数をカウントする事で電力パルスを抽出する様にした。一定時間のパルス数で電力に換算出来るが、この処理はマイコン側、受信アプリ側双方で可能である。

尚、計測結果は、設置済みのパルスカウンターによる電力デマンド装置の出力と比較し、ほぼ一致している事が確認できた。

3.4 データベースの変更

データ蓄積には、データベースとして SQLite を使用しており、このデータアクセス過程において、専用ドライバー(フリーで入手可能)が必要であった。

ここで、企業によっては、Micro Soft 以外のソフトの使用を制限している所もあり、ドライバーソフトとは言えネットワーク内 PC への導入を制限されていた。そこで、ドライバーソフトを必要としない、単純な CSV 形式ファイルをデータベースとして使用する事にした。

特に、本システムでは 1 日毎で新たなファイル名で生成しており、モジュールを約 30 個導入頂いた企業においても数 MB/ファイル程度と比較的小さく、受信データを追加書き込みで処理する事にした。

この結果、書き込みトラブルもなく、処理の高速化も図られている。

尚、従来のデータベースも使える様にデータベース形式は選択可能とした。(図 7)

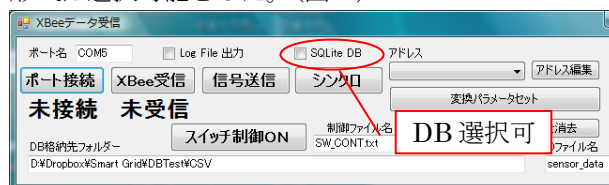


図 7 受信アプリの初期画面

3.5 グラフ表示機能

従来は、受信データのグラフ表示は Excel にてグラフ化するのみで有ったが、状況確認の目的で簡易的にグラフ化したいとのニーズもあり、受信プログラムにてモジュール毎のデータをグラフ化するメニューを追加した。

これは、受信アプリを VB.NET で開発した為、予め開発ツールに用意されているグラフ化フォームを活用して作成した。(図 8)

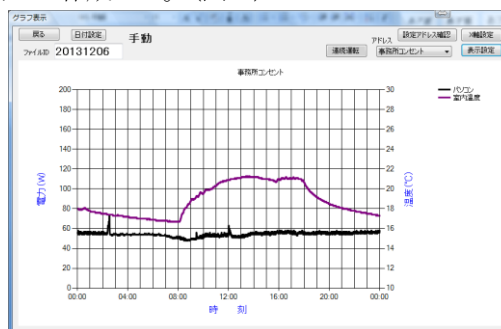


図 8 グラフ表示例

単一モジュールの確認であれば、本グラフのみで

監視出来、一部の企業では本グラフのみで運用し電力削減実績が得られた実例もある。

3.6 データベース確認

企業のニーズではないが、アプリケーションチェック、通信トラブルのチェック等で、データベースの内容を確認する場面もあり、データベース確認メニューも追加した。もっとも、データベースが CSV の場合には直接ファイルの内容を確認する事は容易であるが、SQLite の場合には専用のアプリが必要となるため、デバッグ用としてデータベース確認メニューを追加した。(図 9)

図 9 データベース表示例

3.7 信号送信機能

前報告書にて、XBee モジュールのデジタル端子を ON-OFF させる事でソリッドステートリレーを介して 100V 電源制御が可能となる事を検証済みであるが、今回の受信アプリケーションにも XBee モジュールの制御部を追加した。

この制御は、指定したアドレスの XBee の P1,P2,D1,D2 それぞれの端子に信号を送るもので、PC 画面から端子を選択し送信する事(図 10)は勿論、予め設定しておいた信号送信制御ファイル(図 11)を読みここのファイルの内容に従って、指定アドレスの指定端子を ON-OFF する事も出来る。



図 10 信号送信画面

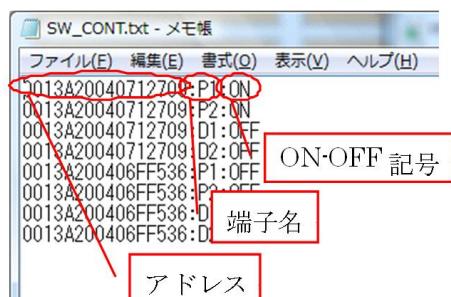


図 11 信号送信制御ファイル

このファイルは、クラウド上に設定する事も可能であり、クラウド上のファイルを外部から修正する事で、物理的には外部から機器の ON-OFF を制御する事が出来るが、無人で機器を操作する事に対する安全面の検討が必要と思われる。

3.8 シンクロ表示機能

受信アプリケーションでは、受信したデータを、決められたパラメータにて電力に換算し、データベースに蓄積する事を基本としているが、例えば電流変化等の比較的早いデータの変化を画面で確認出来るオプションも製作してみた。

企業からのニーズは特に無かったが、様々な動きを確認する時に有効で、図 12 の様に、データベースに出力する事無くグラフのみが表示出来る。

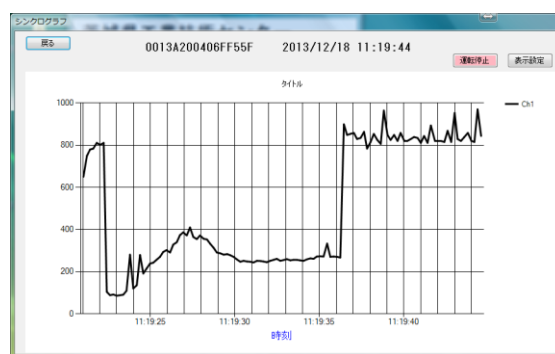


図 12 シンクログラフ表示例

尚、今回製作した、モジュールの処理では、①交流波形を読み、②実行値に換算し、③パケットにセットし、④送信する手順を PIC 内で処理しておりこの処理工程から、最速でも 3 点/秒程度のデータしか送信出来なかった。

PIC のプログラム改善で若干の高速化は見込めると思われるが、本システムでデータ送信レートはこの程度が限界と考える。

3.9 携帯端末での表示機能

今回の研究にて、スマートフォン、タブレット等の携帯端末での応用も試みた。

今回 Android にて試作したがネットワーク内のデータをアクセスする事がやや困難(技術的には可能と思われる)で有った為、フリーで提供されている①クラウド DB にアクセスし②データを携帯端末に取込み、③そのデータをグラフ化するアプリを試作した。

手動操作による、最新データへの更新操作が必要ではあるが一応の動作確認はできた。(図 13, 14)



図 13 Android でのグラフ表示例（折れ線グラフ）

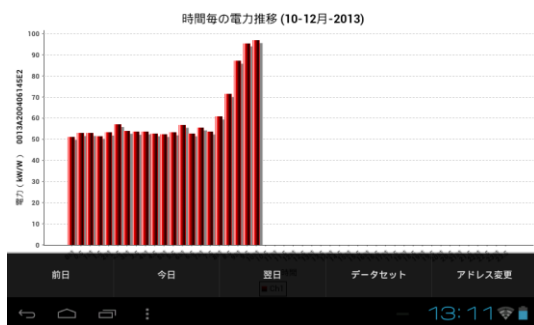


図 14 Android でのグラフ表示例（棒グラフ）

本方式では、アドレス毎の表示となっており、複雑なグラフとなれば、やはり Excel となってしまう。

そこで、別の手法として、①Excel にてグラフ毎のデータを画像 (JPG) データとしてクラウドフォルダーに保存。②保存された画像データをスマホ等からアクセスし、③スマホ等で標準装備されている画像ビューアーで閲覧。(図 1 5)

この手法であれば、既にグラフ化された画像を表示するので、表示には予めクラウドサービスで準備されたアプリで表示でき、クラウドサービスが、iOS, Android 共に提供されていればそのまま表示可能である。

また、グラフの表示内容は Excel で設定するため、様々な表現が可能である。

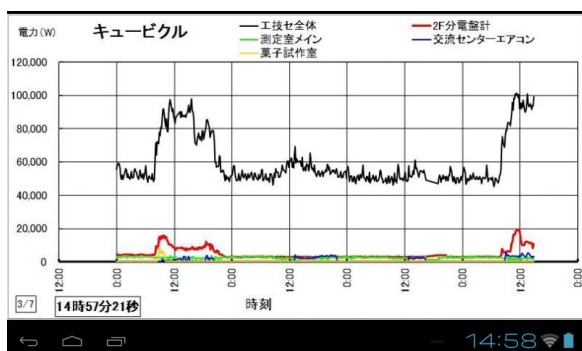


図 15 Android での画像表示例

4. 研究結果と考察

今回の企業での実証実験結果から得られたニーズを元に、有る程度汎用性のある、利便性改善を図ったシステムを構築出来、以下の結果が得られた。

- 1) 装着センサ毎に、モジュール内のプログラムを変更せず対応可能な受信アプリケーションを実現出

来た。また、このセンサ毎のパラメータ設定をビジュアル的に行える様になった。

- 2) データベースを、SQLite と CSV の選択を可能にし、CSV の場合には、データベースアクセスドライバーを必要とせず、運用出来る様になった。また、アプリケーションを VB.NET にて作成する事で Windows PC 専用ではあるが、簡単な操作で動作環境が構築出来るようになった。尚、本システムを運用する、中小企業では Windows PC が普及しており Windows 環境で十分使用可能である事も分かった。

- 3) システムを VB.NET に移行した事により、グラフ表示等の開発ツールが活用出来、①グラフ表示、②データベース表示、③信号送信、④シンクログラフ表示等の周辺アプリを内蔵した利便性の高いツールが作成できた。

- 4) 更にスマートフォン、タブレット等の携帯機器からのアクセスも可能な事が検証できた。今回の実証実験を行った企業からは、まず電力監視がメインのニーズであり、携帯機器からアクセスしたいとのニーズはなかったが、本システムが定着すれば、利便性としての携帯機器からアクセスニーズは出てくるものと思われる。

5. まとめ

今回の実証実験の結果得られたニーズに基づくシステム改良を行ったが、残念ながら現時点で製品化までつなげて頂ける企業は見つからなかった。但し、利用する企業では、電力監視のみならず、生産工程での応用も検討している様で、ハード、ソフトの使用ニーズがある事は判った。この様な企業に対しては、回路図、ソフト、マニュアルを提供しており、企業自身で改善と共に、システム更なる進化を推進して頂けると幸いである。

6. 今後の課題

今回の研究で、ユーザーニーズを反映した、アプリケーションができ、実用も可能と考える。しかしながら、今後改良に企業自身で実施して行くにはやや、困難もあり、ハード、ソフト含めた継続的なサポートを頂ける企業の登場を期待する。

尚、当センターとしても、『技術相談』等で企業お悩みに対する解決策のご相談を引き続き行ってまいりますので、お気軽にご相談下さい。

7. 謝辞

本調査研究にあたり、業務多忙にも関わらず、実証実験にご協力頂き、有意義なご助言をいただいた企業の皆様に改めてお礼を申し上げます。