

Linux マイコンを用いた小型で安価な屋外設置型 太陽光発電遠隔監視システムの開発と 電気電子工学教育への応用

柴田 幸司[†]・若沢 卓道^{††}・花田 一磨^{†††}・関 秀廣^{††††}

Development of an Ultra-small Cost-effective Outdoor Setting-type Solar Power Generation Remote Monitoring System with Linux Microcomputer-based Operation and Related Application to Education in Electrical Engineering

Kouji SHIBATA[†], Takumichi WAKASAWA^{††}, Kazuma HANADA^{†††} and Hidehiro SEKI^{††††}

ABSTRACT

In this study, the authors constructed a system for the acquisition of detailed real-time and time-series data on the power status of solar power generation equipment in sensor-remote locations. The system was created by expanding on an ultra-small Linux-operated remote monitoring system previously developed by the authors. The microcomputer was set to enable graphic representation of values relating to solar panel power generation, battery charge/discharge and load power consumption using an HTTP server, sensor modules and longitudinal information from voltage/current/power sensors. The amount of power generated by solar cell panels, remaining battery capacity and the power consumption of related electrical devices and similar were then remotely monitored using the Internet and a VPN. The results confirmed the system's suitability for remotely monitoring the basic electrical performance of solar power generation facilities. The electrical power generation and energy consumption of the equipment and related items were also monitored over an extended period with the system installed outdoors in a waterproof/weather-resistant case along with a battery and an outdoor waterproof solar cell. The results indicated that the proposed system could be used outdoors without being affected by rough weather. The relationship between the system's power consumption and the solar cell panel/battery used was also studied.

Key Words: *IoT, Internet, VPN, Cellular network, remote monitoring, Solar power, Raspberry Pi, embedded Linux*

キーワード: IoT, インターネット, VPN, 携帯電話網, 遠隔監視, 太陽光発電, ラズベリーパイ, 組み込みLinux

平成 29 年 1 月 6 日受付

† 工学部電気電子システム学科・准教授

†† 工学部電気電子システム学科・4年

††† 工学部電気電子システム学科・講師

†††† 工学部電気電子システム学科・教授

1. はじめに

近年、インターネットや情報通信技術の普及により、マイコンを実装したセンサ機器同士が LAN や公衆ネットワークを介し直接情報のやり

取りを行うセンサネットワークが注目されている^{1,2)}。これらはIoTやM2Mとも呼ばれ、家庭や工場だけでなく植物園や家庭菜園での水分管理、農業や漁業や交通、GPS情報と組み合わせた船舶への設置や車載による移動体などからの各種情報の発信、ヘルスケア、観光業への応用など、ビッグデータ解析などと連携した幅広い用途への展開が期待されている。この様な背景にて、筆者らはセンサ類とインターネットのインターフェースにLinux OSが動作するマイコンを用い、VPNプログラムを組み込みセンサ機器を直接接続し、VPNルータやセンサ情報取得装置を不要とした極めて汎用性が高く、超小型かつ安価で運用コストの低い遠隔監視システムを構築した^{3,7)}。しかし、これらを屋外に設置して安定的に動作する農業や漁業用データの取得や、防災用のテレメータシステムなどを構築するためには、自然エネルギーによる発電電力やバッテリーでの電力消費量などを常に把握する必要がある。そこで本研究では、筆者らが開発したLinuxマイコンを用いた遠隔監視システムを発展させ、従来は発電量の記録を手作業で行っていた⁸⁾太陽光発電設備の発電量などを遠隔地から自動的に把握するため、マイコンに接続したセンサから電圧・電流・電流の時系列データを取り込みhttpサーバで公開し、インターネットおよびVPN経由で遠隔地にて取得する太陽光発電遠隔監視システムを文献⁹⁾や文献¹⁰⁾を参考に構築した。さらに、システム一式を耐候ケースに入れ屋外に設置し、同じく屋外に設置した太陽光パネルと接続して発電量やバッテリーでの装置の消費電力等を長期にわたり観測し、本システムが屋外にて長期間使用できることを実証すると共に、得られた結果から遠隔監視システム自身の消費電力に対する、太陽光パネルやバッテリーの選定法の関係などの検討を行った。

2. システム概要

今回構築したシステムは図1および図2の通り、

制御用のマイコン¹¹⁾と電圧・電流センサモジュールから構成される。システムに用いた図3の電流・電圧・電力センサには、マイコンとI2Cのインターフェースにて通信が可能で複数のアドレスから選択が可能なTexas Instruments社の1チップ電圧・電流センサICであるINA226を用いた。このICは業界最高クラスの精度を持つ電流シャント・モニタICで16ビットのA/Dコンバータを内蔵しており、変換時間は最高140 μ sである。また、電流計のシャント抵抗には0.002 Ω の値を用いたため回路への影響は非常に小さく、配線ケーブルの抵抗に比べても無視できる大きさである。測定可能な電圧は直流36V、電流は双方向で ± 20 Aまでとなっている。また、電流は双方向で測定可能であるため、太陽光パネルから電池へと電流が流れ込むだけでなく、電池から負荷へと流れ出すようなバッテリーの充放電現象の監視にも利用できる。さらに、モジュールに取り付けたシャント抵抗値が極めて低いことから、10Aを超える電流をレンジの切り替えなしにmAの分解能にて測定できる。これにより、本測定システムはテスターのように電流レンジによって内部抵抗が大きくなったりしない。また検出回路の電流センスはハイサイド・ローサイドどちらにも対応でき、低抵抗で高精度・温度係数が小さい電流センス抵抗を採用している。これにより、オフセット10 μ V、ゲイン誤差0.1%とクラス最高の測定精度を実現している。

構築したシステムは、マイコンとセンサ間の通信にシリアルデータの転送が容易に実現可能なI2Cインターフェースを用い、センサチップとマイコンとの結線はそれぞれ、センサのSCLおよびSDAはマイコンのSCL (5番) およびSDA (3番) ピンと接続した。さらに、センサのVCCおよびGNDは、それぞれマイコンの3.3V (1番) およびGND (6番) と接続した。なお、I2Cインターフェースを有するセンサとマイコンとの接続は、モジュール基板間を空中配線により行った。配線状態は図1に示す通り、ICモジュールや抵抗などの電子部品が効率よく配置できている。



図1 システムの写真

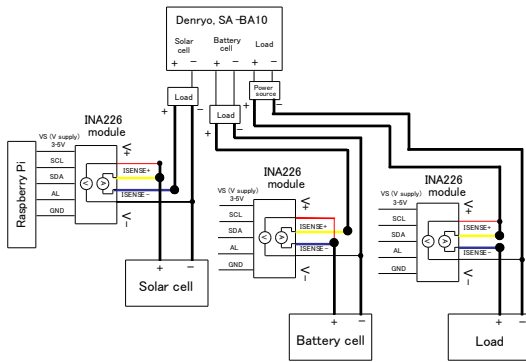


図2 システムの回路図

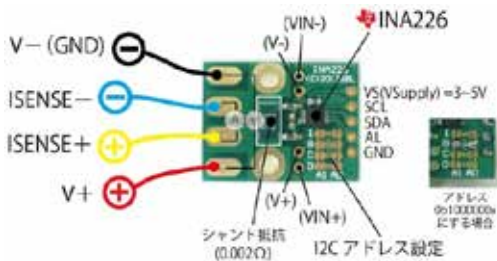


図3 電圧・電流データモジュールの配線

その際、今回のケースではINA226 I2Cデジタル電流・電圧・電力計モジュールを複数配置して全てのモジュールをラズベリーパイと通信させるため、各モジュールは個別のI2Cアドレスとすることが必要。I2C (Inter-Integrated Circuit)とは、主に同じ回路またはチップ間で通信する用途のシリアル通信方式である。そこで、3つのアドレスを表1に示す通り40h, 41h, 42hとして設定し、

それぞれのアドレスに対応するようランドの半田付けを行うことにより、マイコンと各モジュールへのシリアル通信を実現している。

表1 I2Cアドレスとランドの関係

I2Cアドレス	A1	A0
40	G-A1	G-A0
41	G-A1	1-A0
42	G-A1	D-A0

3. I2C通信の有効化

先の手順にてラズベリーパイに I2C の端子を介しセンサの接続が完了後、I2C デバイスとの通信を有効にさせるため

sudo nano /etc/modules

と入力して設定ファイルを開き、「i2c-dev」なる記述にてラズベリーパイが認識できるデバイス名を追加した。また、Raspberry Pi の module のデフォルトの設定では「I2C」および「SPI」通信はブラックリストにされているため

sudo nano /etc/modprobe.d/raspi-blacklist.conf

でブラックリストの設定ファイルを呼び出し「i2c-bcm2708」の行を「#」によりコメントアウトしてブラックリストから外し、「Ctrl+o」で上書き保存「Ctrl+x」で終了した。

次に i2c-tools のインストールを行う。I2C は知れる通信での通信規格であり、SCL (シリアル・クロック) と、双方向の SDA (シリアル・データ) の 2 本の信号線 (GND は含まない) で通信する。また、バスには複数のスレーブを接続でき、マスタは個別に決められたスレーブのアドレスを指定してスレーブを選択し、そのスレーブと通信を開始する。また、I2C がサポートするビットレートには標準モード、ファースト・モード、ハイスピード・モードがある。今回は I2C インターフェースによるセンサモジュールからの電流・電圧およびデータの取り込みのため、まず最初に以下の 2 つのコマンドを入力して I2C および Python に関連するソフトウェアをインストールした。

```
sudo apt-get install i2c-tools
sudo apt-get install python-smbus
```

その際、「raspi-config」の「Advanced option」にてI2Cを有効化する（非常に重要）。これにより、I2Cでの通信が可能となったので、デバイスアドレスの確認をするため

```
「sudo i2cdetect -y 1」
```

と入力した結果、図4のように「0x40」、「0x41」および「0x42」として、3つの電流センサモジュールであるINA226を同時に認識した。さらに、センサとマイコンがI2Cを介して正しく通信されていることを確認するため、試しに

```
「sudo i2cget -y 1 0x40 0x00 w」
```

と入力すると、同じく図4に示す通り、センサから「0x2741」という返答があり、マイコンとセンサ間で正しく通信が出来ていることが確認できた。

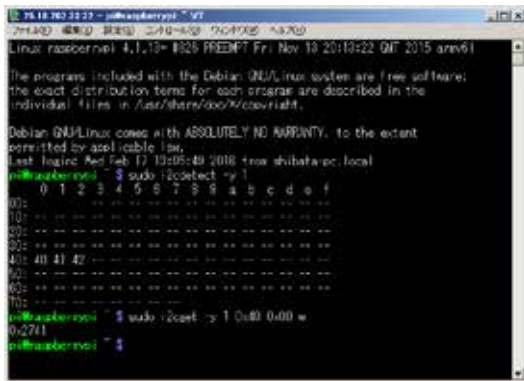


図4 デバイスアドレスと通信の確認

4. 電圧・電流および電力データの取り込み

ここで、INA226にて情報取得するための各レジスタのアドレスは表2の通りであり、これらのアドレスを入力すれば各コマンドが実行される。

表2 INA226におけるレジスタアドレス

ポインターアドレス	レジスタ名	初期値	Read or Write
00h	コンフィグレーションレジスタ	01000001 00100111	R/W
01h	シャント抵抗電圧レジスタ	00000000	R
02h	バス電圧レジスタ	00000000	R
03h	電力レジスタ	00000000	R
04h	電流レジスタ	00000000	R
05h	キャリブレーションレジスタ	00000000	R/W
06h	マスク/イネーブルレジスタ		R/W
07h	アラートリミットレジスタ		R/W
FFh	IDレジスタ		R

ここでまず、「Calibration レジスタ」に適切な値を設定すると後述の「Current レジスタ」と「Power レジスタ」が有効になる。そのための設定値は「 $2.5 \times 2048 \div \text{シャント抵抗値}[\text{m}\Omega]$ 」で求まる。これより、シャント抵抗の値が 0.002Ω の時はレジスタに記憶させる設置値は 2560 となるので、回路の正しい電気特性を取得するためには、キャリブレーションレジスタ 05h には 10 進数 2560 の 16 進数である 0A00h をセットすれば良い。その為に

```
sudo i2cset -y 1 0x40 0x5a 0x000a w
```

を実行する。そののち、センサモジュールで検知しているシャント抵抗間の電圧を

```
sudo i2cget 1 0x40 0x01 w
```

というコマンドで読み出したところ、「0x4201」という値が得られた。次に、バス電圧の値を

```
sudo i2cget 1 0x40 0x02 w
```

にて読み出したところ、「0xb927」という値が得られた。ここで、この 16 進数の値の最初の 1 桁とその後の 2 桁を入れ替えると、27B9h という値は 10 進数では 10169 となり、この値を 1.25 倍したものがシャント抵抗の端子間電圧[mV]となる。よって、 $10.169 \times 1.25 = 12.711\text{V}$ となる。ここで、オームの法則 $I=E/R$ より $R=0.002\Omega$ 時のバッテリー電流は $I=3.55 \times 10^4 / 0.002 = 0.177\text{A}$ となる。一方、電力レジスタは

```
sudo i2cget 1 0x40 0x03 w
```

の入力にて「0xc600」が得られ、電流レジスタは

さらに、各機器における電力の時系列データも自動取得することを目的として、文献¹⁰⁾を参考にスクリプトファイルを作成した。具体的にはまず、エディタにて

```
sudo nano /home/pi/get_ina226panel0a.php
```

のファイルを開き図8のphpファイルを記述して「Ctrl+o」と「Ctrl+x」にて上書き保存し終了した。

```
<?php
//ina226 データの取得

// Calibrationレジスタに a00h を書き込み
exec("sudo i2cset -y 1 0x40 0x05 0x000a w");

$vol = exec("sudo i2cget -y 1 0x40 0x02 w"); // 電圧レジスタ読み込み

$vol = hexdec(substr($vol,-2).substr($vol,2,2)) * 1.25 / 1000;

printf($vol);
?>
```

図8 記述したスクリプトファイルの内容

但し、root ユーザでなければ生成したスクリプトファイルを実行できないため

```
sudo chmod u+x get_ina226panel0a.php
```

として実行権限を与えた後

```
sudo chmod 774 /home/pi/get_ina226panel0a.php
```

でアクセス権限を拡大し、コマンドラインより

```
php /home/pi/get_ina226panel0a.php
```

と入力したところ、図9の通り 11.34V なる値が出力された。



図9 スクリプトファイルによる電圧データの自動取得

6. 各観測点の電圧・電力の時系列データ取得

そこで、電圧・電流および電力センサのINA226にて観測された値をMUNINで読みここれらの時系列グラフを作成するため、図9のスクリプトを記述し“/usr/share/munin/plugins/solar”なるファイル名として配置して“chmod 775 solar”にてアクセス権限を変更した。

```
#!/bin/bash

#%# family=auto
#%# capabilities=autoconf

available="yes"
case $1 in
config)
echo "graph_title Solar Power Generation Voltage/Power"
echo "graph_category solar"
echo "graph_vlabel Voltage(V)/Power(W)"
echo "graph_args -l 0 -base 1000"
echo "voltage.label Voltage"
echo "voltage.draw LINE2"
echo "power.label Power"
echo "power.draw LINE2"
exit 0
;;
autoconf)
if [ "$available" = "yes" ]; then
echo "yes"
exit 0
else
echo "no (daemon isn't running)"
exit 1
fi
;;
snmpconf[suggest])
exit 0
;;
*)
;;
esac
#データの読み出し
SOLAR=php /home/pi/get_ina226panel0a.php
SOLAR1=php /home/pi/get_ina226panel0b.php
echo "voltage.value $$SOLAR";
echo "power.value $$SOLAR1";
```

図10 発電量取得のためのMuninスクリプト

そして

```
sudo ln -s /usr/share/munin/plugins/solar /etc/munin/plugins/solar
```

としてシンボリックリンクを作成し、Muninがroot権限で値を取得できるように

```
sudo nano /etc/munin/plugin-conf.d/munin-node
```

のMuninの設定ファイルに

```
[solar]
user root

[solar1]
user root

[solar2]
user root
```

を追加して該当プロセスを一度killしたのち“sudo munin-node restart”した。

これらのソフトウェアの設定の後、装置一式を

図 11 の通り八戸市にある大学の屋内に設置し、さらに図 12 に示すように太陽光パネルは屋上に配置して、装置と防水処理を施しつつ接続し、2015 年 12 月末から 1 か月以上にわたり、装置の連続稼働時における発電量や装置での消費電力等の長期観測を行った。



図 11 電気棟 4 階でのシステムの稼働の様子



図 12 電気棟屋上への太陽光パネルの設置

その結果、図 13 および図 14 の通りパネル、蓄電池および負荷の電圧および電力の時系列データが取得できた。まず、図 13 では 2015 年 12 月 29 日(火)と 30 日(水)の 2 日間の発電量の時間変化について、瞬時的には 40W 以上の発電が確認出来るものの晴天時と曇天時の発電量には大きな違いが見られる。また、この期間の夜間の無発電時も含めた平均発電電力が 3.3W ということは、八戸地域で 50W の太陽光パネルを用いた場合は常時運用可能な機器の消費電力が概ね 3.0W 以下

となる必要があり、装置の動作の維持がかなり厳しいことが確認できる。

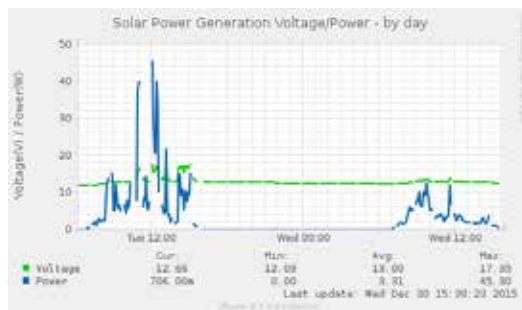


図 13 2 日間にわたるパネル電圧と発電電力の変化

一方、2 日間の負荷出力は図 14 の通り、このラズベリーパイを用いた太陽光発電遠隔監視システム自身の消費電力は 3W 程度と一定である一方、負荷両端の電圧は太陽光パネルによる発電と出力負荷によるバッテリー電力の消費の影響により、17V から 12V 程度まで変化している。但し、2015 年 12 月 28 日から 2016 年 1 月 5 日までの間の八戸市における太陽パネルによる発電量と発電電圧の推移は図 15 の通りで、この期間に十分な発電量が得られたのは 2015 年 12 月 31 日と 2016 年 1 月 5 日の 2 日のみであった。

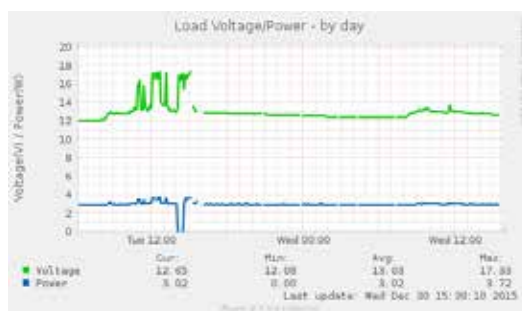


図 14 2 日間にわたる負荷の電圧と電力の変化

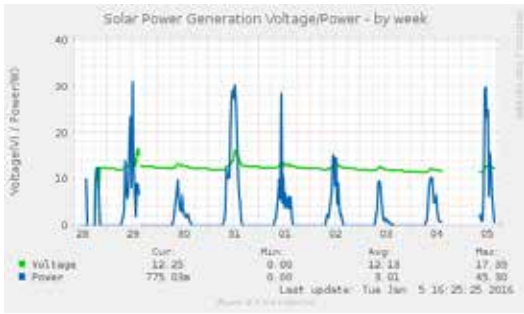


図 15 1週間にわたる太陽光による発電量の実績

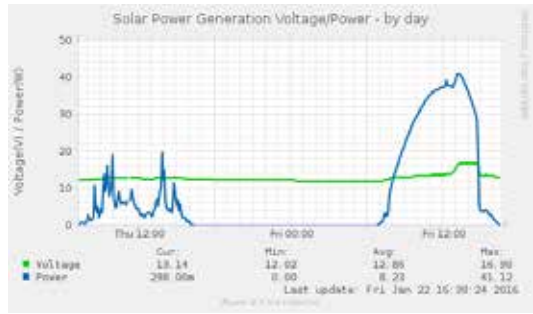


図 16 晴天時の発電特性

しかし、その後の1月22日（金）には2016年に入って最大の発電量を記録した。図 16 に示すように前日との発電量の差は明白で、連続して35Wを超える時間が3時間を越えている。このことから、その他の時間の発電量の積算も考慮すれば $35 \times 4 = 140\text{Wh}$ 以上は発電していると思われる。但し、実際の1日のトータルとしての発電量は各時間の発電量の瞬時値の時間積分（弧の面積）で計算できるので、正確な発電量はこの理論に基づいたシステムへのプログラムの組み込みが必要である。今日の発電特性はデータが綺麗な弧を描いているので、この時期のチャンピオンデータと考えることが出来る。また、この図より15時以降に急激に発電量が低下していることも確認でき、この時間から日が西に傾きパネルへの入射角が大きくなることが理由と思われる。さらに、1ヶ月近くの観測で分かってきたことは、冬場には数日間にわたり発電が見込めない日があるため、晴天の日に蓄えた電力を長時間消費するために、より大容量のバッテリーが必要なことである。以上のことから、八戸市での太陽光パネルを用いた発電特性については大分把握が出来てきたので、引き続き長期間にわたりシステム改善のためのデータ収集を続け、八戸市における冬場の太陽光発電の有効性につき検討の必要がある。

7. 屋外設置の実績と製造コスト

本システムは八戸市という冬場の寒さが厳しい地域での屋外での使用を想定した。そこで、システム一式を防水耐候ケースに取り付け屋外設置し、同じく屋外に設置した太陽光パネルと共に1ヶ月以上にわたる運用試験を実施した。その様子は図 17-19 に示す。この結果、図 20 の通り途中で大雪に見舞われたがシステムは常に動作している。そして図 21 の通り、半年間におよぶ連続運転が確認でき、雪天・雨天や常時間の温度サイクル変化に耐えて運用できることを確認した。但し、7月、8月の晴天時には庫内の温度上昇により複数回マイコンがハングアップしたので、今後は庫内の冷却対策を万全にすると共に、より長期間にわたる運用試験を実施して特に夏場の気温が上昇時も含め実績を重ねる。



図 17 晴天時のシステムの稼動の様子



図 18 雪中でのシステムの稼動の様子



図 19 大雪時のシステムの稼動の様子

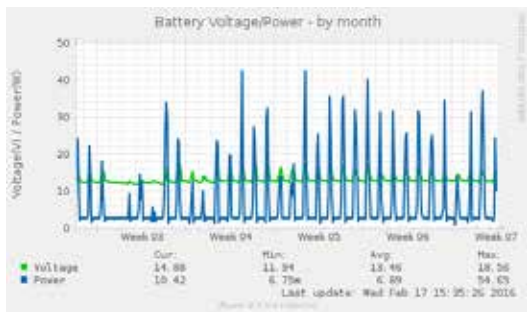


図 20 一ヶ月間にわたるシステムの稼動状況

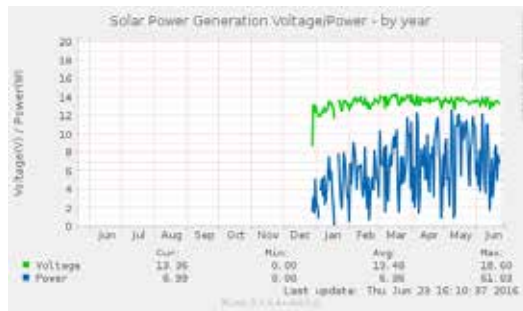


図 21 半年間にわたるシステムの稼動状況

この太陽光発電遠隔監視システムの構築に用いた主な機材や部品および価格を表 4 に示す。合計金額は 1 台あたり 45000 円程度であり、太陽光パネルおよびバッテリーが全体価格に大きな負担となっていることが分かる。一方、本システムを太陽光発電状況の確認教材という単一目的と考えると、現状のシステムが常時消費する電力である 3W をより省電力化すれば、太陽光パネルやバッテリーに負担をかけることが少なくなり、より安価な太陽光パネルやバッテリーを選択できる。よって、より消費電力の小さいマイコンやマイコン周辺機器を採用したシステムの省電力化が今後の検討課題である。

表 4 用いた部品リスト

品名	型番	数量	価格 [円]
ARM-Linux マイコン	Raspberry Pi 1 B+	1	3200
Wi-Wi 子機	WLI-UC-GNM2		1500
TEXAS INSTRUMENTS 電流・電圧センサモジュール	INA226	3	1200
バッテリー (12V,12Ah)	WP22-12NE	1	7000
太陽光パネル	AT-MA50A	1	15000
充放電コントローラ	SA-BA10	1	5000
太陽光パネル接続ケーブル	2SQ : H-CV(MC4)	1	4000
TAKACHI 防水ケース	BCAP303018T	1	5000
TAKACHI 取付ベース	BMP3030W	1	1500
TAKACHI 外部取付足	CK-26P or BFL-2	1	500
ミラフレキ-SS	MFS-16 (50 巻)	1	500
PF 管コネクタ	MFSK-16G (10 入)	1	300
組端子台 (6 端子)	T10-06PM	1	300

8. まとめ

本報告では Linux マイコンを用いた組込み型 VPN により、超小型で安価で運用コストの低い太陽光発電遠隔監視システムを構築した。そして、遠隔地の発電およびバッテリーの充放電情報を取得するための電流・電圧および電力データの取り込みおよび公開法について検討した。その結果、I2C を利用することによりマイコンに接続されたセンサモジュールにて電圧・電流および電圧データの長期間にわたる時系列情報が容易に遠隔監視できることを確認した。さらに、システム一式を耐候ケースに入れて長期運用試験を実施し、冬場の大雪にも耐え連続稼働できることを確認した。

今後はマイコンや周辺機器の選定を見直すことによる省電力化や、これに伴う太陽光パネルやバッテリーの小容量化の可能性の検討を行い、市内の公共施設や観光地等に設置する予定である。また、提案システムを用いた大人数の大学生に対する装置の製作およびプログラミング実習などによる電気電子工学教育としての展開を行うと共に、たとえば市内の小中学校等に配布して理科教育や環境問題の教育に役立てることが課題である。また、温度・湿度および日照センサなどと組み合わせ、これらと発電量との関係についても検討を行う。さらに、引き続き屋外設置による連続運転試験を行い、夏場など炎天下での長期運用が可能かも実証が必要である。

参考文献

- 1) Zhen Zhu and Ruchun Cui, "Remote Intelligent Monitoring System Based on Embedded Internet Technology," Proceedings of the 2007 IEEE International Conference on Automation and Logistics, pp. 2665-2669, 2007-8.
- 2) Y. Ha, "Dynamic Integration of Zigbee Home Networks into Home Gateways Using OSGi Service Registry," IEEE Transactions on Consumer Electronics, vol. 55, no.2, 2009.
- 3) 柴田幸司, 花田一磨, 落合翼 "Linux マイコンを用いた組込み VPN による超小型センサ情報遠隔監視システムの開発" 八戸工業大学紀要 33, pp115-120, 2014-3.
- 4) 柴田幸司, 花田一磨, 飯野真弘, 武 美里, 赤塚一磨 "Linux マイコンを用いた組込み VPN による超小型センサ情報遠隔監視システムの開発と教育への応用" 信学技報 教育工学研究会, Vol.114, No.441, ET2014-83, 2015-1.
- 5) 柴田幸司, 飯野真弘, 武 美里, 赤塚優磨, 花田一磨 "震災対応のための Linux マイコンを用いた超小型センサ情報遠隔監視システムの開発とネットワーク教育への適用," 電子情報通信学会総合大会, D-15-5, 2015-3.
- 6) 成田博貴, 菊地桐吾, 柴田幸司 "Linux マイコンによる安価な超小型センサ情報遠隔監視システムの開発とネットワーク教育への応用," 2015 年度電気関係学会東北支部連合大会, ID01, 2015-8.
- 7) K. Shibata and K. Hanada "Development of an Ultra-small Sensor Information Remote Monitoring System with an Embedded VPN and Linux Microcomputer Operation", Proceedings of International Conference on Engineering and Applied Science, ICEAS2015, Sapporo, Japan, 2015-7.
- 8) 柴田幸司, 武 美里, 花田一磨, "遠隔監視システムの屋外常時運用のための太陽光発電と蓄電池による独立電源の評価 (第一報)" 八戸工業大学紀要 34, pp101-107, 2014-3.
- 9) "太陽光発電の電力計をラズベリーパイで作ってみた" <http://myboom.mkch.net/modules/pukiwiki/178.html>
- 10) "気圧センサ BMP085 を RaspberryPi に接続しグラフ表示" <http://blog.bnikka.com/raspberrypi/bmp085raspberrypi.html>
- 11) Raspberry Pi ホームページ <http://www.raspberrypi.org/>

要 旨

本研究では、筆者らが以前開発したLinuxマイコンを用いた超小型遠隔監視システムを発展させ、マイコンに接続した電圧・電流・電力センサからの値を時系列データとして取り込み、遠隔地に設置した太陽光発電設備の太陽光パネルによる発電、バッテリーの充放電、さらには負荷での消費電力などの情報をhttpサーバからグラフとして公開してインターネットおよびVPNを介し、リアルタイムかつ時系列に取得するシステムを構築した。その結果、今回設計および製作したI2Cインターフェースおよび電圧・電流センサモジュールを用いた回路により、太陽光発電設備の電気性能の基本となる電流、電圧および電力データの長期間にわたる時系列情報が容易に遠隔地にて監視できることを確認した。さらに、バッテリーも含めたシステム一式を耐候ケースに入れ屋外に設置し、同じく屋外に設置した太陽光パネルと防水処理を施して接続し、発電量やバッテリーでの装置の消費電力などを長期にわたり観測した。この結果をふまえ、本システムが悪天候に影響されず屋外で使用が可能であることを実証すると共に、遠隔監視システム自体の消費電力と用いる太陽光パネルやバッテリーの選定法との関係についても検討を行った。

キーワード：IoT, インターネット, VPN, 携帯電話網, 遠隔監視, 太陽光発電, ラズベリーパイ, 組込みLinux