

市販マイコンによるセンサ情報の収集とインターネット回線による遠隔でのモニタリングに関する一手法

A Study on Remote Monitoring System via Internet Communication Using Commercial Microcomputer

朴 忠植* 金子 憲一**
Park Choong Sik Kaneko Kenichi

(2016年7月21日 受理)

キーワード：モニタリング, インターネット, マイコン

1. はじめに

小型、高精度、高機能なセンサやマイコンの低コスト化が進み、企業だけでなく個人でも、容易にモニタリングや機器制御システムの試作が可能になってきている。また、これら電子機器とインターネット技術をリンクさせ、IoT(アイオーティ Internet of Things)のキーワードで人と物、物と物に関連する情報を処理するツールを提供するIT企業が出現している。これらのサービスの中には、単なる情報を蓄積管理するだけでなく、統計処理や機械学習によるデータの特徴や予測に関する分析ツールも提供しているところもある。工業分野では、自社の開発機器のデータ解析や工場内で使用する加工機器などの故障予測にツールを活用し、高品質なものづくりや生産工程の効率化に役立てようとしている企業も増えてきている¹⁾。

当研究所の自動化、計測制御分野を担当する部署では、マイコンやPCを用いた計測制御システムの試作を行ってきた。近年では、大学との共同研究で垂直軸型風力発電に関するインターネット回線を介したモニタリングシステムの開発を行った。本報告では、遠隔でのモニタリングシステムを構成する基本的な技術要素の種類と特徴を述べ、誰でも入手可能で安価な市販のマイコンおよびフリーウェアのソフトウェアを多用したモニタリングシステムの試作事例について紹介する。

2. システム技術要素の種類と特徴

リモートモニタリングシステムの機能は、要約すれば、対象物の状態に関する情報を各種センサで計測し、計測データを保管し、離れた場所のユーザーに計測データの換算値やグラフで認識しやすくディスプレイ画面で表示することである。また、異常検出時にスピーカーやランプ等で通報し、場合によっては、対象物の動作を緊急停止させることである。

計測対象とモニタリング場所が近い場合は、RS232CやGPIBなどのレガシーな通信方式により計測機器とPCを接続し、専用ソフトウェアによってデータ表示・計測機器操作を行う構成となるが、RS232Cの最大ケーブル長は15mと短く、また、同方式の長距離用に拡張したRS422Aは1200mではあるが、障害物を越えての敷設は困難である。一方、インターネットはLTE(Long Term Evolution, 携帯電話の通信規格)、Wi-Fi(ワイファイ, Wireless Fidelity, 無線方式の通信規格)、イーサネット(有線方式の通信規格)など複数の方式で通信インフラとして、至る所で利用が可能である。IPアドレスによる通信相手特定する世界共通の通信プロトコルで随時インターネットに情報機器を接続すれば、世界中のコンピュータ上の情報を瞬時に入手できる。そのため、リモートモニタリングシステムの開発には、計測対象と観測する場所の距離や数の制約の少ないインターネットによるデータ通信は有効な手段であるといえる。インターネット回線を介

* 制御・電子材料科

** 大阪府立大学大学院工学研究科

表 1 主な市販マイコンボードの種類と仕様

| 品名 仕様 | Arduino UNO | mBed LPC1768 | Raspberry Pi | Intel Edison Kit for Arduino |
|----------|------------------------|----------------------|-----------------|----------------------------------|
| CPU | 16MHz 8bit | 96MHz 32bit | 700MHz 32bit | 500MHz 32bit 2 コア |
| メインメモリ | 2K | 32KB | 512GB | 1GB |
| 入出力端子 | デジタル入出力 14 アナログ入力 6 | アナログ入力 6 アナログ出力 1 | デジタル入出力 40 | デジタル入出力 20 アナログ入力 6 |
| 無線通信 | 外付け | 外付け | 外付け | Bluetooth, Wi-Fi |
| OS | 無し | 無し | Linux | Linux |
| プログラム言語 | 専用言語 | 専用言語 | C, C++, Java など | C, C++, Java など Arduino 用専用言語 |

してモニタリングシステムを構成するには、センサや計測制御用マイコン、通信デバイス、アプリケーションが基本的な技術要素となる。以下、各構成技術の利用上の注意点を中心に説明する。

2.1 センサ

センサは計測対象の温度や圧力などの物理量や濃度や成分などの化学量を電圧信号に変換して出力する電子部品である。正電圧、正負電圧、パルス電圧や出力範囲が数 mV 程度から数十 V など、出力の仕様はセンサの種類やメーカーによって異なり、センサの出力を処理する計測制御用マイコンの入力仕様の範囲内にあるか注意する必要がある。

マイコンと専用ケーブルで接続し、直接デジタル値を出力する 3 軸加速度センサなどの高機能センサもあるが、マイコン側の接続数に限りがあるので多用はできない。

2.2 計測制御用マイコン

プログラムによりサンプリング周期や各センサ出力の特性やバラツキに応じた処理などを柔軟に調整することができる。国内外メーカーから多くの種類のマイコン用 IC が市販されているが、マイコン選定には、用いるセンサの数や種類に応じた仕様であるか確認が必要である。マイコン用 IC では、マイクロチップ社の PIC が種類も多く、技術情報もインターネット上に豊富にあるので、国内では試作品、製品ともよく使われている IC である。産技研でも PIC を用いた技術支援も多く、PIC の応用開発のレポート²⁾も発行しているので参照されたい。

PIC 自体は IC なので、センサなど周辺の機器と繋ぐのは面倒であるが、基板上に PIC とともに入出力用コネクタや USB やシリアル通信、イーサネット通信機能を有したマイコンボードがあるので、システム試

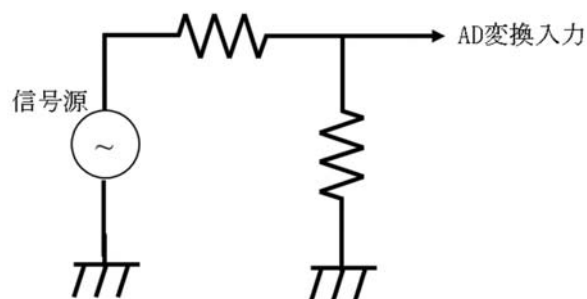


図 1 抵抗分圧回路

作には便利である。表 1 に主な市販のマイコンボードの種類と仕様を示す。いずれも店頭やネットで入手が容易であり、入出力信号の種類（アナログ、デジタル、シリアル通信）、数、コストが選定の際の基準となる。マイコンに関する技術の進化も目覚しく、今後、さらに高性能で低コストなものが利用できるようになるであろう。

マイコンの多くは正信号のみで、かつ最大電圧および入出力電流量も制限がある。センサ出力をそのままマイコンに入力できない場合には、電圧変換等の回路を付加する必要がある。図 1 に基本的な電圧変換回路である抵抗分圧回路を示す。

プログラム言語には、汎用言語のアセンブラや C 言語が利用可能である。また、容易に利用できるように簡潔な書式で処理を記述できるプログラム言語とプログラミングに必要なエディタやコンパイラ、サンプルプログラム等の開発環境をインターネットから無償でダウンロードして利用できるものも多い。

2.3 通信プロトコル

インターネット回線を通じてデータの送受信を行うには、データフォーマットおよび手順を規定する通信プロトコルに従ってプログラムを作成する必要

がある。通信プロトコルを表2に示す。データに乗せた電気信号とPCやマイコンで処理するデジタル信号との相互変換を司る再下位層のネットワークインターフェース層から、メール通信やWebなどの各種情報処理に応じたプロトコルを定義する上位層のアプリケーション層まで、4つの階層で区分することができる。

アプリケーション層のプログラムの実行には、トランスポート層のプログラムが呼び出され、トランスポート層のプログラムの実行には、インターネット層のプログラムが呼び出されるような流れとなる。ユーザレベルで情報通信を行うプログラムはアプリケーション層に属する。この層の代表的なプロトコルであるWebサーバとWebブラウザ間の通信プロトコルHTTP(Hiper Text Transfer Protocol)やファイル転送用プロトコルFTP(File Transfer Protocol)は、送受信の要求や送受信確認などの処理により、通信の安定性が重視されている。

表2 インターネットワーク階層

| | | | |
|----|-----------------|------------------------------|-------------------------------|
| 4層 | アプリケーション層 | ユーザーが利用するアプリケーションに関する | HTTP,SMTP FTP,MQTT CoAP |
| 3層 | トランスポート層 | データの送受信確認等の手順を規定する通信の信頼性に関する | TCP,UDP |
| 2層 | インターネット層 | 送受信先の設定に関する | IP,ARP |
| 1層 | ネットワークインターフェース層 | 物理的な通信方式に関する | イーサネット Wi-Fi |

クライアント サーバ クライアント サーバ

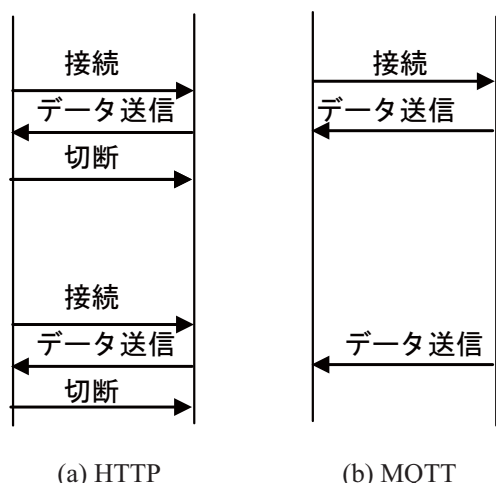


図2 HTTP(a)とMQTT(b)の通信シーケンス

アプリケーション層のMQTT(Message Queue Telemetry Transport)やCoAP(コープ:Constrained Application Protocol)は、M2M(Machine to Machine)通信すなわち機械間のリアルタイムのネットワーク通信を目的に作られたプロトコルである。IoTブームの高まりを受け、最近になり、IBMやAmazonを筆頭に、これらのプロトコルを採用した自社のサーバやクラウドとユーザー側のセンサ機器をつなぐツールを提供する企業が増えている。自前で構築するシステムより自由度は低いが、短期間でシステムを構築できるので、このようなツールの活用も有効である。

MQTTやCoAPはHTTPに比べて軽いプロトコルと表現される。通信データにおいて通信手順を簡略化、すなわち本来のデータの前に付加される、通信のやり取りに用いられるデータ(ヘッダ)を小さくすることにより、送信元と先でのデータ処理時間を短縮している。HTTPのヘッダは最小50バイトに比べて、MQTTのヘッダは最小2バイトである。

図2にHTTPとMQTTとの通信手順の概念的なシーケンスを示す(実際は、サーバとクライアント間で場合によって異なる処理となる)。HTTPではデータの送信毎に接続と切断を行うが、MQTTでは、接続後に明示的な切断がなければ、接続の状態を維持しデータが発生した時点で送信が行われるので、HTTPに比べてオーバーヘッドが少なく、より低消費電力でリアルタイムな通信が可能となる。比較的新しいプロトコルなので技術情報も少なく、HTTPやFTPに比べて、導入するには時間がかかるかもしれない。

2.4 無線方式

センサ側からサーバへデータを転送するには、センサの配置や数の変更が容易になるので、有線より無線が有効である。無線方式の種類と仕様を表3に示す。

仕様は最大値を表しており、通信速度は公称値、他は、およその値である。ZigBee(ジグビー)やWi-SUN(Wireless Smart Utility Network)は、65000以上の対象を同時計測することができ、低消費電力なので多数の対象の計測には向いている。

表3 主な無線方式の種類と仕様

| 種類 仕様 | Wi-Fi | Wi-SUN | Bluetooth | ZigBee |
|----------|---------|---------|-----------|---------|
| 通信速度 | 6.4Gbps | 400kbps | 24Mbps | 250kbps |
| 通信距離 | 100m | 500m | 100m | 70m |
| 接続数 | 32 | 65535 | 7 | 65535 |
| 消費電力 | 1 W | 1mW | 100mW | 10mW |

Wi-Fi 方式以外はインターネット通信プロトコルと異なるため、インターネット経由でサーバと通信するには、他の通信方式でマイコンや PC 上にデータを収集した後、インターネット通信のプロトコル変換をしてサーバに転送するような方法が必要となる。

2.5 データ管理

計測制御用マイコンからサーバにデータを転送する場合に、データ列かファイル形式のいずれかの方法をとることになる。

データ列の場合は、サーバ側でデータを受け取った後、所定のファイルに追記保存するプログラムを事前に作成し、実行させておく必要がある。民間の IoT サービスを用いれば、サーバ側の処理はファイル名などパラメータの設定だけで済むようになっている。

ファイル形式の場合は、HTTP や FTP を利用できるが、マイコン側でサーバからの応答メッセージの解析、判定などの煩雑な処理が必要となり、CPU パワーの小さいマイコンには適していない。3章の事例では、計測制御用マイコンとサーバ間に、データ列からファイル形式に変換後、サーバに再送するための PC レベルの性能を持つゲートウェイ装置を設置し、FTP によりデータ転送を実現している。

ユーザーへのモニタリングデータの表示は、Web サーバ方式が適している。モニタリング用ソフトをパソコン毎にインストールすることなく、随時、最新の計測データを Web ブラウザで表示でき、同時に複数の場所からモニタリングが行える。また、PHP や JavaScript などのプログラム言語を併用すれば、グラフ処理用のライブラリを利用できる。さらに、サイトの情報にアクセスできる API(Application Programming Interface) をプログラム中に用いて、地図データや気象データ提供サービスサイトと連携させ、多面的に情報のモニタリングも可能である。

3. システム試作事例

3.1 試作目的

風力発電に関する研究用として試作し、大阪府立大学内に設置したモニタリングシステムの事例を紹介する。風力発電は風車の形状により水平および垂直軸型に分けられる。一般的に水平軸型が垂直軸型より発電効率が高く、普及化が進んでいる。一方、垂直軸型は水平軸型と比べて設置・メンテナンスの容易さと風向きに対する依存性がなく、また騒音が小さいということで、人家のある場所にも利用可能であり、太陽電池などと併用した小規模な発電装置を各所に設けて、局

表 4 センサの出力仕様

| センサ名 | 出力仕様 |
|-------------|----------|
| アナログ式風向・風速計 | 0V~1V |
| トルク計 | -10V~10V |
| 回転数計 | 0V~10V |

地でのエネルギーをまかなう分散型エネルギー源に適していると考えられる。

筆者らは、金属板などのリジッドな材質を用いた従来の導風板を樹脂などの板に置き換えることで、風力に応じて形状が変形するフレキシブルな導風機構 FGV (Flexible Guide Vane) に関する研究開発を行っている³⁾。従来、強風による過剰な回転速度防止のためにある速度以上 (cut out) では強制停止させていた風速域においても、FGV により風の一部を風車に呼び込み、過回転にならない程度で回転させて運転時間を延ばすことで、低い発電効率を補うことが可能であると考えられる。

研究当初は、大学単独で屋外での風の変化に対する機構の動作特性および発電効果を調査するために、風向・風速センサによる風の状態と回転数・トルクセンサによる計測データを実験装置の傍に設置したデータロガーで収集した後、USB メモリを介して、他の場所にある PC でデータ処理を行っていたために作業効率が悪かった。そこで、計測を遠隔で行うモニタリングシステムの開発を当所が行った。

3.2 システム要求仕様

本研究の目的以外のモニタリングツールとして利用できるように、汎用性のあるシステムについて検討し、仕様を以下のように設定した。

(1) 表 4 に示すセンサを用いる。

(2) 風力発電装置のある実験場所から他の建物内の研究室の PC に計測データを保管するサーバを設置する。利用者らが EXCEL 等で解析処理を行えるよう CSV 形式で保存する。

(3) サーバには HTTP 機能を持たせ、別の PC、タブレットなどの情報端末でリアルタイムにモニタリングをする。

(4) 運転状況を画像としてモニタリングする。

(5) センサの種類、個数の変更に対応する。

3.3 システム構成

前節の要求仕様に対して、以下に示す 5 つのモジュール構成とした。

(1) センサ計測装置

マイコンを用いて、センサデータの収集の後、デー

タ変換をし、ゲートウェイ装置にデータ送信を行う。マイコンに Wi-Fi 方式のモジュールを付加し、TCP 通信を行う。マイコンには、Arduino Uno(アールデュイノウノ)を用い、無線モジュールには Wi-Fi 通信が可能な ZigBee モジュールを用いた。

(2) ゲートウェイ装置

センサ計測装置から受信したデータを CSV 形式のデータファイルに変換し、サーバに転送する。装置には Linux OS を搭載したマイコン Raspberry Pi(ラズベリーパイ)を用いた。単一プログラムしか処理できないマイコンに比べて、複数のプログラムを同時に動かすマルチタスク処理ができる。

Linux OS は、OS の機能を利用できるコマンドが豊富にあり、スクリプトプログラム(テキスト形式のコマンドを組み合わせたプログラム)により複雑な処理を行うことができる。スクリプトプログラムは、CPU の仕様に合わせて機械語に変換するコンパイルの必要がなく、処理動作のテスト確認、修正が迅速に行えるので、試作には便利である。

センサ計測装置との通信には、トランスポート層に属する TCP プロトコルを用いた。図 3 に Python 言語で記述したセンサ計測装置からの受信処理のスクリプトプログラムを示す。接続要求や接続承認、接続開始、送信、終了などの処理単位に対応した機能を持つソケット関数を用いている。同じローカルネットワーク内には、センサ計測装置以外に通信する装置がなく、ネットワーク上のトラフィックがほとんどないため、送信エラー時の確認、再送手順などを省略した。通信トラフィックによるエラー処理が必要な場合は、TCP プロトコルで煩雑な処理をプログラムで記述するより、HTTP などを利用すればよい。

計測時間の管理は、センサ計測装置側でネットワーク経由により時計サーバから得た時刻を用いることを考えたが、単体プログラム中でのセンサ計測と時刻取得の同時処理は時間の遅れが生じる。ゲートウェイ装置上で動作する Linux OS のほうが時刻合わせもマルチタスク処理で行え、CPU パワーも PC 並みなので、より正確な時刻を得ることができる。

そこで、ゲートウェイ装置側で、センサ計測装置から受信したデータを CSV ファイルに変換する際に、受信時刻を計測時間として追記することにした。計測直後にデータを Wi-Fi 経由で送信し、ゲートウェイ装置側は秒以下の周期で受信データの有無を確認処理している。同じ Wi-Fi ルータ下でのデータ通信なので、電波上の遅れは無視できるほどであり、計測時間と実際の計測時間との時間差は、通信プロトコル上の処理

◆ ホストの IP アドレスと通信ポートの設定

```
host = '192.168.11.3'
port = 10002
x=0
```

◆ TCP 通信開始

```
serversock =
socket.socket(socket.AF_INET,socket.SOCK_STREAM)
serversock.setsockopt(socket.SOL_SOCKET,
socket.SO_REUSEADDR, 1)
serversock.bind((host,port))
while True:
    serversock.listen(1)
    clientsock, client_address = serversock.accept()
    x=x+1
```

◆ データ受信時の処理

```
while True:
    rcvmsg = clientsock.recv(1024)
    if rcvmsg != "":
        f = open("wind_monitor_data.txt","w")
        f.write(rcvmsg)
        f.write("\n")
        f.close()
    clientsock.close()
    y=str(x)
    os.system('sh ./wind_monitor_sub.sh '+y)
    break
```

図 3 センサ受信処理のプログラム例

時間のみで数秒程度であり、今回のモニタリングでは問題ないレベルである。

(3) Wi-Fi ルータ

実験場所に学内 LAN と繋がる Wi-Fi ルータを設置し、(1)と(2)の間の通信は Wi-Fi ルータのもとでネットワーク管理される LAN 構成とした。(1)と(2)は、ローカルなアドレスで管理し、Wi-Fi 自体に、データサーバと通信可能な、学内ネットワーク下で管理される固定アドレスを設定した。

(4) モニタリングサーバ

ゲートウェイ装置から送られる CSV 形式のデータファイルを受信する FTP サーバと Web ブラウザでの

モニタリング画面を表示する Web サーバを用いた。サーバ用 PC は Windows7 が搭載されていたので、無償利用可能な Web, FTP サーバなど各種サーバ機能を持つ I2I サーバを採用した。グラフ処理表示にはプラグイン Dygraph(ダイグラフ)を用いた。

(5) 画像モニタリング

ゲートウェイ装置に接続した USB カメラをモニタリングに用いた。画像保存と画像ファイル転送処理はスクリプトプログラムで記述した。動画撮影は、Raspberry Pi では負担が大きく動画が途中で停止した

りする場合があったので、センサデータをサーバに送信する際に、同時に撮影した静止画を送信することにした。画像撮影には、USB カメラ操作ライブラリである fswebcam(エフエスウェブカム)を用いた。

システム構成におけるセンサデータの具体的な流れを図4に示す。センサ計測装置とゲートウェイ装置、WiFi ルータを含む実験装置は大学建物の屋上に設置され、別の建物内にモニタリングサーバを設置した。実験装置の写真を図5に示す。

図6に学内LANに接続可能な別のPCのブラウザ

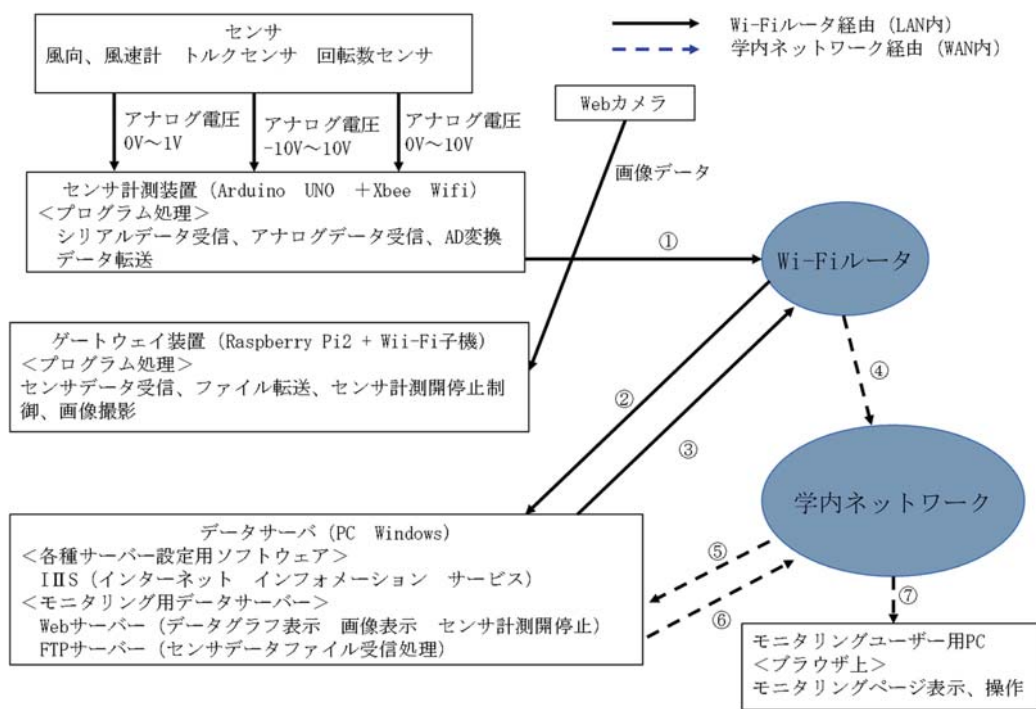


図4 通信データの流れ

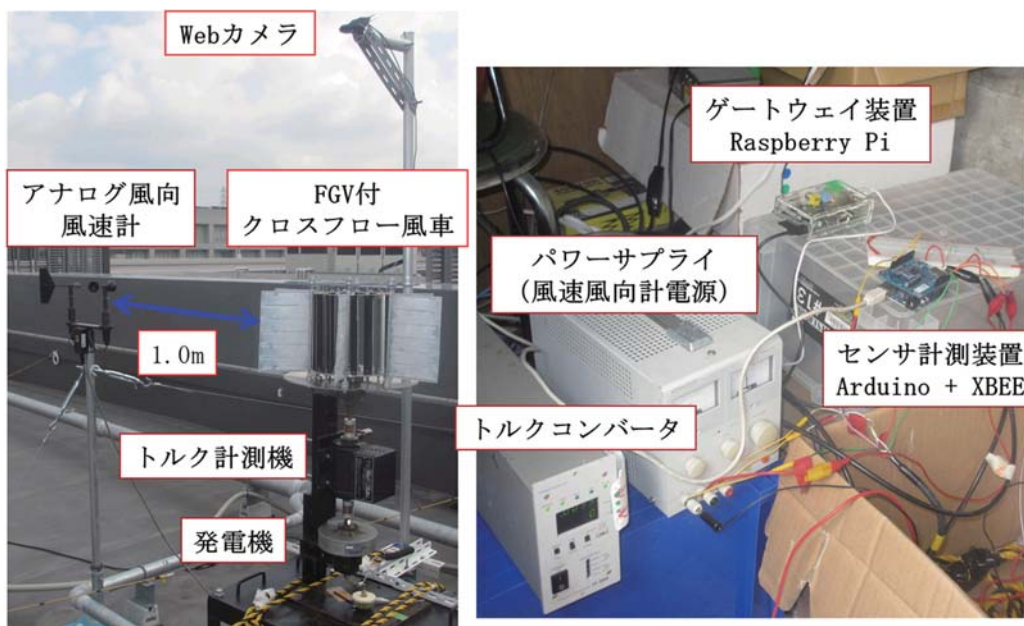


図5 実験装置および実験システム

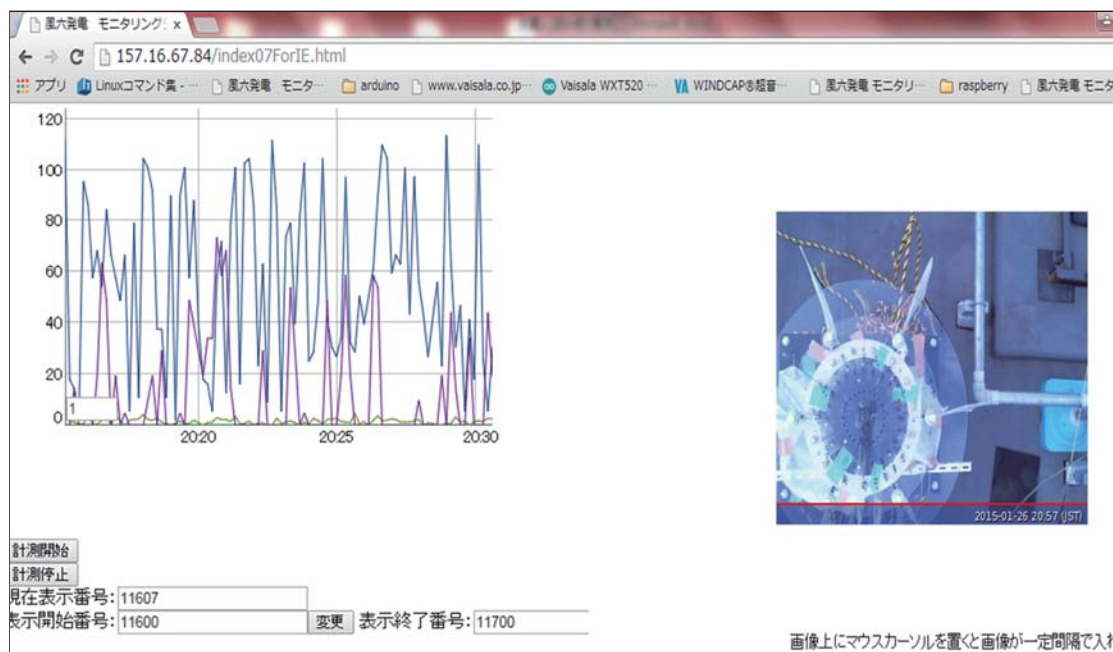


図 6 Web ブラウザ上のモニタリング表示

から見たモニタリング表示画面の一例を示す。10分おきに数日間の風速等のデータが表示されており、自動で更新されるようになっている。また、Dygraph の持つ機能によりマウスで表示範囲の変更などが行える。表示画面下部に、計測開始と停止ボタンを設けている。これは、センサデータのモニタリングと逆方向の流れで、開・停止を表すデータをサーバからゲートウェイ装置、そしてセンサ計測装置へと送受信処理するプログラムを各装置に記述することにより、計測動作の操作を実現している。このように、単にモニタリングだけでなくモニタリング対象への操作も可能となっており、風力発電装置の制御装置が外部入力で操作可能であれば、緊急時の停止や再起動も可能となる。

4. おわりに

市販マイコンを用いたインターネット回線によるモニタリングシステムの構築に関して、主な技術要素と

して、ハードウェアおよびソフトウェアならびに通信方式の種類と特徴について説明し、風力発電装置用モニタリングシステムの試作事例を紹介した。

高機能、高性能なハードウェアやソフトウェアが安価に入手でき、また、インターネットサイトの豊富な技術情報や詳細なシステム化事例を参考にして、アイデアを容易に実現できるようになっている。

本報告が自社の実務や実験ツールとして、モニタリングシステムを導入する際の参考になれば幸いである。

参考文献

- 1) 松元則雄, 今井拓司: 日経エレクトロニクス 4月号, (2016) 49.
- 2) 北川貴宏, 朴 忠植, 谷口正弘: 大阪府立産業技術総合研究所報告, No.27 (2013) 13.
- 3) 金子憲一, 岡田 怜: 日本マリンエンジニアリング学会誌, 49, 2 (2014) 194.