RO-013

クラウドコンピューティングによる遠方監視制御システムの試作
A Prototype of SCADA System using Cloud Computing

宮西 洋太郎†
Yohtaro Miyanishi

1. はじめに

近年，情報システムの実現形態として，Web アプリケーションの形態，さらにはクラウドコンピューティングの形態が注目され，普及しつつある．電子情報通信学会 SWIM 研究専門委員会では，2010 年度筆者らの呼びかけにより，クラウドコンピューティングについて，集団的に取り組むプロジェクト（CCTP: Cloud Computing Trial Project）を立ち上げ，筆者を含め多くの参加者がクラウド技術の試行と修得を行った[1][2][3][4]．このプロジェクトは 2010 年度に終了したが，引き続き有志メンバーが活動を発展的に継続している．

プロジェクトで取り組んだ情報システムは，外部からのデータ入力や指示は http プロトコルによるという人間系に閉じる一般的なシステムであったが，今回，筆者はクラウドコンピューティングの一応用として，機械系（machine）と人間系（man）を統合するシステム，すなわち M2M（Machine to Man/Machine）システムとしての遠方監視制御システム(SCADA: Supervisory Control And Data Acquisition System)を試作した．機械系はセンサデータをマイコン（マイクロプロセッサ）によって収集し，そのデータをクラウドコンピューティング環境にあるサーバマシンにアップロードするという形態である．基本的な目標（ディジタル入力, アナログ入力の監視機能の実現性検証）は達成できた．制御機能（ディジタル出力）については，今後の課題であるが，実現の可能性が見えてきた．

この試作を通して得られた知見と課題について報告する．

2. 開発の背景

CCTP では，参加者から試作・検証すべき各種アプリケーションの提案が募集された．筆者もいくつかの提案を行ったが，そのうちのひとつに「クラウドと M2M の連接システム」があった（2010/3/31）[5].

その提案の概略は，

(1) センサユニット（センサ＋マイコン）が市販されているとして，それを用意する（このユニット用のネットワークがすでに用意されているとして）．
(2) センサユニットを USB でパソコン（以下 PC）に接続し，使いたいサービスを設定する．
(3) センサユニットを PC から切り離し，センサを動作させたいところに設置する．
(4) あとは, PC からセンサユニットからの情報を見るだけ．

という提案であった．すなわち，緊急かつ突発的なセンサデータ収集のニーズに対して，できる限り迅速に対応できるシステムを目指している．この提案は，残念ながら CCTP の試作システムとしては，採用されなかったが，筆者は興味を持ち続けていたので試作することにした．

同様な主旨の既存システムをインターネットで調べたところ，原発事故などに用いることを想定したセンサネットワークシステムの存在があった[6]．筆者が試作したシステムは，例えば，留守宅や別荘の遠方監視や老人の見守りなどの目的に，できるだけ準備が少なくて，実現できる方法の確立を目指すものである．また，今後のセンサを扱う多様なシステムの開発のためには，自らの技術修得として，試作することが必要であり，試作した．また準備を少なく（できれば準備不要に）するため，サーバはクラウド環境を用いることとした（クラウド利用の大きな理由）．

マイコンとクラウド環境（今回は GAE: Google App Engine）との間の結合の仲介装置として PC を用いた．マイコン側のプログラミング, 及び GAE 側のプログラミングについては，従来と同様であるが，今回新たに得られた知見として，マイコンと PC の間は「シリアル通信」で，PC とクラウド環境との間は「Http クライアント」の方法で，Java 言語によってプログラミングを行い実現したことを以下に述べる．

3. 遠方監視制御システム

電力系統などのように，地理的に広範囲に配置された監視制御対象を遠隔にあるセンターから監視制御する方法として，図 1 に示すように，従来から CDT（Cyclic Digital Telemeter）や TC(Tele-Control)といった専用のデータ伝送装置がインターネット普及以前から用いられている．これらの装置は工業用途として高い信頼性，高速な応答性が要求され，高額な装置として実現されている．

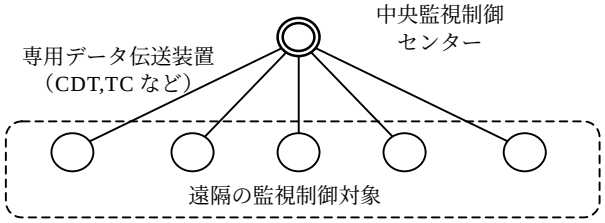

図 1 従来型の遠方監視システム

このような用途に対して，現在のインターネット技術（TCP/IP，Web 技術，普及した光・電気ネットワーク）とマイクロコンピュータ技術を用いれば，信頼性，応答性の問題は将来の課題としつつも，従来に比べ，きわめて低価格で同様の機能を実現できる可能性がでてきた．図 2 に構成を示す．

図 2 において，遠隔の監視制御対象は現場での小規模コンピュータ（すなわちマイコン）によってデータが収集され，インターネットを通じて，中央におけるサーバにデータが送り込まれる．サーバは中央監視制御センターと同じサイトまたは独立の別のサイトに設置される．

インターネット利用のタイプでは，付随的なメリットとして，中央監視制御センターからだけではなく，インターネットに接続された任意のサイト・端末から対象の監視制

†㈱アイエスイーエム　ISEM, Inc.

御を行うこともできる．例えばスマートフォンを用いれば，通勤途中でも遠隔の対象の監視制御を行うことができる．

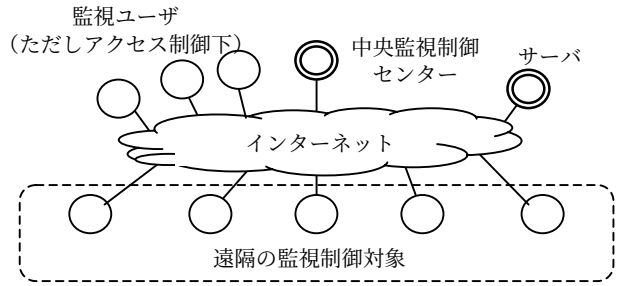

図 2 インターネット利用遠方監視システム

インターネットを通じて，遠隔の監視対象からデータがサーバに集められる．今回の試作では，遠隔の監視対象をマイコンによってデータ収集し，サーバとしては，一般的なサーバあるいはレンタルサーバとすることもできるが，今回は，前述の理由もあり，クラウドコンピューティング環境下のサーバとした．

4. 試作システム

4.1 試作の目的

試作の目的は，クラウドコンピューティングによる遠方監視制御システムの実現性を確認することである．すなわち監視制御対象からセンサデータを自動的に収集し，クラウドコンピューティング環境下にあるサーバにデータを集約し，ユーザが遠方監視制御を実現できることを確認することが目的である．

また別の観点から見ると，従来の CCTP において，人間系に閉じる情報システムの実現については確認済みであった．それに対して，センサデータのような機械系のデータを連携させることが，どのような方法で可能であるのかを見出し，システムを動作させることによってそれを確認することが目的である．

今回の試作は，監視機能に絞った．制御機能については，中央からの直接的な制御は，IP アドレスの関係で，おそらく IPv6 の普及を待つ必要があると考えている．ただ，後述するように実験の過程において，直接的な制御ではなく，制御する方法も見えてきた．

4.2 システム構成

大枠においては，図 2 のシステム構成をとるが，遠隔のマイクロプロセッサからサーバへのデータ伝送の方法として，次の 3 種類のタイプを想定した．

・タイプ A
センサ→マイコン→PC→インターネット
→サーバ（クラウド環境下）
・タイプ B
センサ→マイコン→有線 LAN→インターネット
→サーバ（クラウド環境下）
・タイプ C
センサ→マイコン→無線 LAN／携帯電話網
→インターネット→サーバ（クラウド環境下）

4.2.1 タイプ A

図 3 にタイプ A のシステム構成を示す．本テーマの実現性を確認するためには，最も基本的な構成であり， 今回の実験は，このタイプを試作し，動作の確認を行った．

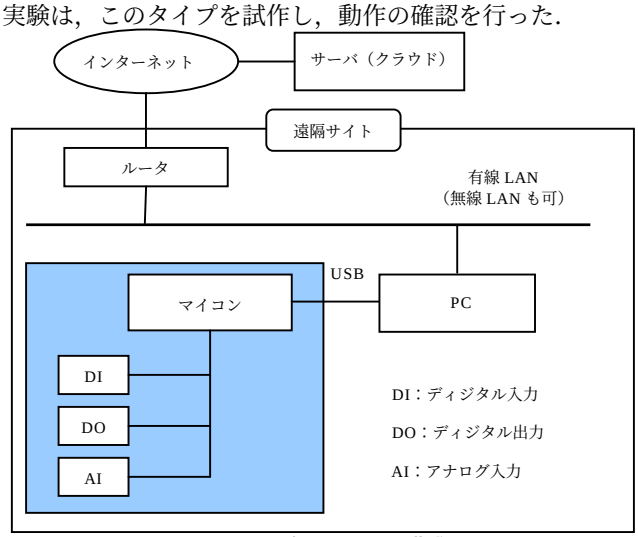

図 3 タイプ A システム構成

マイコンは，センサからのプロセス入出力 (PIO: Process Input and Output) データを入出力（今回は入力のみ）し，USB シリアル通信で PC に送信する．

PC は準備期間中ではマイコンプログラムの開発環境を担当し，運転開始前にはマイコンへの指示を担当し，運転中ではマイコンからデータ受信し，Http クライアントとして，データをサーバに送信する役割を担当する．

4.2.2 タイプ B

図 4 にタイプ B のシステム構成を示す．タイプ A の確認の後に，開発に着手する予定である．

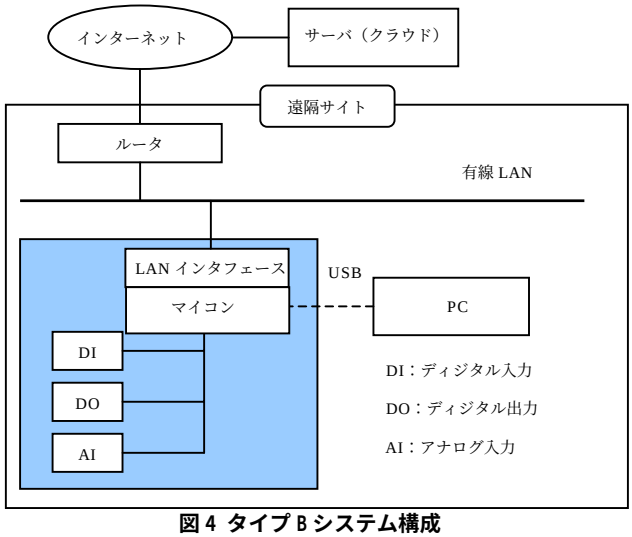

図 4 タイプ B システム構成

図 4 において PC は，マイコンプログラムの開発環境及びマイコンに対して運転開始前には各種指示をおこなうために用いる．運転時には，PC を切り離しても，マイコンから LAN 経由で，サーバにデータを送ることが可能である．

4.2.3 タイプ C

図 5 にタイプ C のシステム構成を示す．タイプ B の確認の後に，開発に着手する予定である．

図 5 において，タイプ B と同様に，PC は，マイコンプログラムの開発環境及びマイコンに対して運転開始前には各種指示をおこなうために用いる．運転時には，PC を切り離しても，マイコンから無線環境経由でサーバにデータを送ることが可能である．

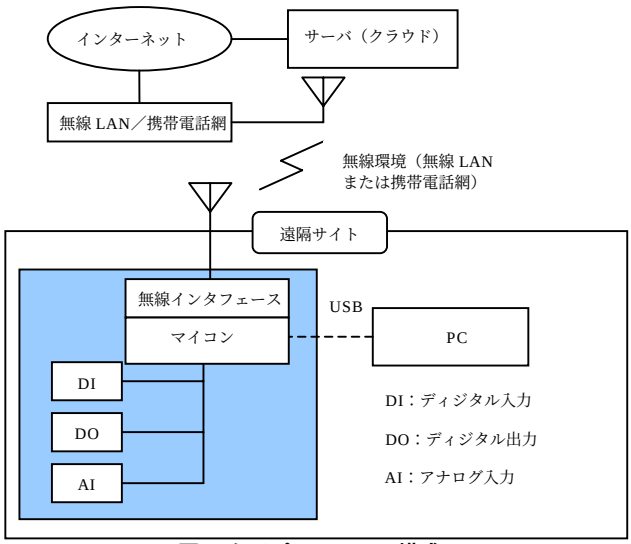

図 5 タイプ C システム構成

このタイプでは，無線環境が準備されている場所では，ケーブル接続もなく，遠隔サイトに電源を内臓させれば，単純に遠隔サイトを持ち込むのみ，すなわち前述のセンサユニットを用意するのみで，遠方監視動作が可能となる．

4.3 システムの機能

4.3.1 アプリケーション範囲

・このシステムに複数のアプリケーションを含む．（例：留守宅監視，別荘監視，老人見守り監視，室温監視，農場監視など）
・１つのアプリケーションに複数のノード（マイコン）を含む．（1 つのマイコンに接続する PIO 点数に上限があるため，ノード数を必要に応じて増設する．）ネットワークにおけるマイコンをノードと称することにする（図 3,4,5 における「遠隔サイト」に同じ）．
・１つのノードに複数の DI(Digital Input), DO(Digital Output), AI(Analog Input)を含む（今回の実験は，DI および AI が対象）．各種のセンサ（例：リミットスイッチ，温度，湿度，使用中電力，風量）を DI 及び AI として接続し，データを取り込む．

4.3.2 マイコンの機能

・各種機能について，PC からの指示（あるいは事前の設定）を受け取る．
・指定の周期（１秒の指定倍数）で DI を行う．
・指定の周期（１秒の指定倍数）で AI を行う．
・1 秒周期で，DI 入力値の状態変化を検出し，変化時に PC にシリアル通信で変化データを送信する．スイッチ接点のチャタリング対応のため，指定周期の入力で，現在状態が 0 の場合，1,1 の連続で 0 から 1 への変化，現在状態が 1 の場合，0,0 の連続で 1 から 0 への変化したと判断する．
・AI データは指定周期で PC にシリアル通信で送信する．
・マイコンでの周期処理は基本的にディレイ処理によるが，追加処理により，１m 秒の単位での精度をもたせる．

4.3.3 PC の機能

・各種機能について，マイコンに指示を与える．
・マイコンからの受信データをシステムコンソール（System.out.println）に表示する．
・マイコンからの受信データのうち，DI の状態変化及び周期的 AI データを，サーバに送信する．

4.3.4 サーバの機能

・ユーザの指示で，各種の初期設定を行う．
・遠隔サイトの DI 変化をサーバ内のデータストアに格納し，ユーザの要求に応じて，ブラウザに表示する．DI 値は，接点がオン／オフといった生データを，人間にとって意味のある表現に変換する（例えば，「玄関扉：開／閉」の開／閉）．
・遠隔サイトの AI 値（周期的スキャンによって得られた値）をサーバ内のデータストアに格納し，ユーザの要求に応じて，ブラウザに表示する．AI 値は，電圧が何ボルトといった生データを，人間にとって意味のある表現（工学値，EU(Engineering Unit)値）に変換する（例えば，「居間温度：22℃」の温度値）．

図 6 に監視画面の例を示す．破線内が今回実装範囲である．

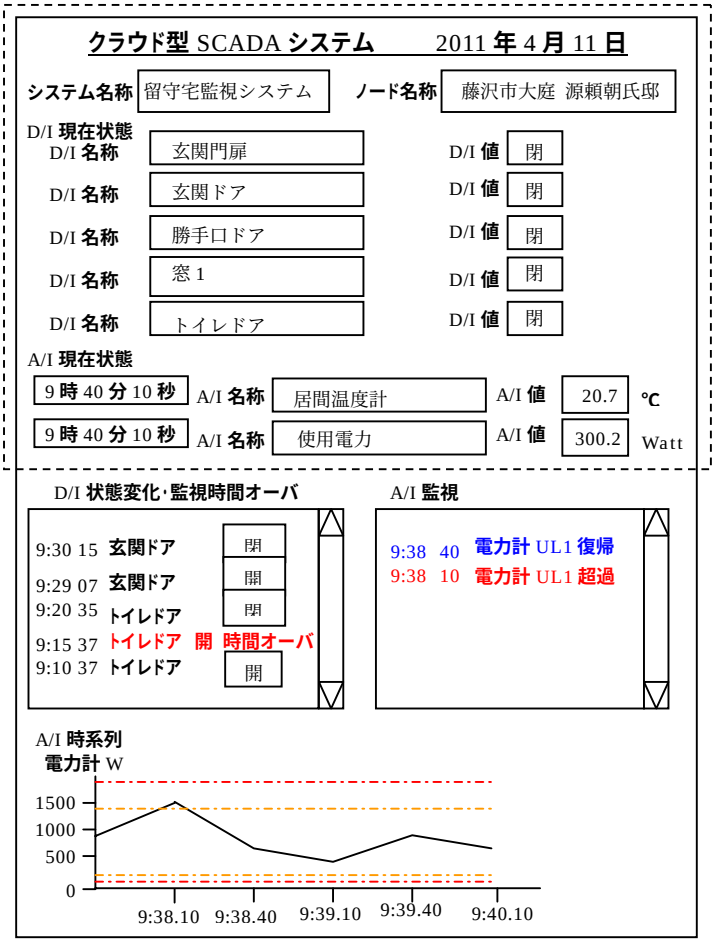

図 6 監視画面の例

4.3.5 PC からマイコンへの送信データ項目

・アプリケーション番号の設定

‘anX’　X：アプリケーション番号（0～9）
・ノード番号の設定
‘nnX’　X：ノード番号（0～9）
・ディジタル出力オン
‘dnX’　X：DO 番号（0～4）
・ディジタル出力オフ
‘dfX’　X：DO 番号（0～4）
・ディジタル入力
‘diX’　X：DI 番号（0～4）
・ディジタル入力全点
‘da’
・ディジタル入力スキャンオン
‘dsX’　X：周期インデックス（0～9）
・ディジタル入力スキャンオフ
‘dsf’
・アナログ入力
‘aiX’　X：AI 番号（0～4）
・アナログ入力全点
‘aa’
・アナログ入力スキャンオン
‘asX’　X：周期インデックス（0～9）
・アナログ入力スキャンオフ
‘asf’
・ディジタル入力変化検知オン
‘cdn’
・ディジタル入力変化検知オフ
‘cdf’

4.3.6 マイコンから PC への送信データ項目

マイコンから PC へ送信されるデータは，PC にとどまる項目と，PC からさらにサーバに送信される項目がある．
(1) PC からサーバに送信しない項目
・PC からマイコンへの指示データのアンサーバック
‘xxx，n=yyy’　xxx 指示データ，yyy ミリ秒
・ディジタル入力スキャンデータ
‘dsX’に対応したデータ
・ディジタル入力全点データ
‘da’に対応したデータ
・アナログ入力全点データ
‘aa’に対応したデータ
・ディジタル出力データ
‘dnX’，’dfX’に対応したデータ
・アナログ入力データ
‘aiX’に対応したデータ
(2) PC からサーバへ送信する項目
・ディジタル入力変化データ
‘cdn’に対応したデータである．
‘a-n-dX’　X は変化後のディジタル入力値（0 または 1）
a-n-d は DI の id（diid），a はアプリケーション番号，n はノード番号，d は DI 番号．
・アナログスキャンデータ
‘ais’に対応したデータである．
‘a-n-aXXXX’　XXXX は 4 桁のアナログ入力値, 4 桁にするためのバイアスとして 1000 を加算している．
a-n-a は AI の id（aiid），左から順に，a はアプリケーション番号，n はノード番号，a は AI 番号．
(3) スキャン dsX，asX のスキャン周期インデックス

10 種類のスキャン周期を指定する．以下の周期はプログラムの定数としてコーディングしているが，変更は可能である．

・X=0：1 秒周期	・X=1：5 秒周期
・X=2：10 秒周期	・X=3：30 秒周期
・X=4：1 分周期	・X=5：5 分周期
・X=6：10 分周期	・X=7：30 分周期
・X=8：1 時間周期	・X=9：2 時間周期

4.4 システムの実装

4.4.1 マイコンの選択

今回の実験では，マイコンはどのような機種でもよく，入手が簡単なことと，マイコンに実装するプログラムが簡単に作成できることがあえての条件であった．そのような条件から，書店で入手できるものを試しに購入してみたところ，プログラミングも C 言語や Java 言語と同等レベルの記述でできることが確認できた[7]．ただし，配列などの記法に問題があったが，配列にせずに個別の変数にすることにより，問題を回避できた．

参考文献[7]のマイコン Japanino は，基本的に Arduino マイコン（マイコンボード）であり，Arduino 用の IDE（Integrated Development Environment）が利用可能である[8]．

また，センサについては，ブレッドボードを利用して，接点オン／オフ DI，LED 点滅 DO，可変抵抗器による AI（0～5Volt）によって，センサを模擬した．

4.4.2 サーバの選択

筆者は，[3]において選択理由も述べ，GoogleAppEngine（GAE）を選択した．ある程度慣れていたので，今回もこれを選択した．ただし，今回は GAE の特徴でもあるスケールアウト特性を用いるまでもない規模の実験であるので，Java が使えるならば，通常のレンタルサーバでも事前に準備をしておけば，実験は可能であると思われる．

4.4.3 クラス設計

今回の試作において，サーバ（GAE）で KVS（Key Value Store）に永続化する主要なクラスの設計は，従来の RDB におけるデータベース設計とほぼ同様であり，主な作業は，あるクラス（RDB におけるスキーマに相当）にどのような属性をもたせるかという設計作業である．それに加えて，付随的に属性のデータ処理（例えば生データから EU 値への変換など）がメソッドとして付け加わることになる．

主なクラスについて紹介する．クラスについて，詳細に述べる理由は，クラスにどのような属性を持たせているかを明らかにすることによって，将来機能も含めて，当該情報システムにどのような機能を具備させているかを明確にできるからである．ただし紙面の都合もあり代表的クラスの記述にとどめる．

GAE 特有の制約として，整数は Long 型，浮動小数点は Double 型のみとなっている．

(1) ノード	RemoteNode
・ノード id	nodeid　String
・アプリケーション番号	applinumber Long
・ノード番号	nodenumber Long
・アプリケーション名称	appliname　String
・ノード名称	nodename　String
・DI 点数	numberofdi Long
・DO 点数	numberofdo Long

・AI 点数　numberofai Long
・DIid0　diid0　String
・・・・
・DIid4　diid4　String
・DOid0　doid0　String
・・・・
・DOid4　doid4　String
・AIid0　aiid0　String
・・・・
・AIid4　aiid4　String
(2) リモート DI　RemoteDI
・DIid　diid　String
(アプリケーション番号，ノード番号，DI 番号を”-“を介し連接)
・アプリケーション番号　applinumber Long　(0～9)
・ノード番号　nodenumber Long　(0～9)
・DI 番号　dinumber　Long　(0～4)
・アプリケーション名称　appliname String
(KVS 対応の非正規化のため)
・ノード名称　nodename String
(KVS 対応の非正規化のため)
・DI 信号名称　diname　String
・DI 今回入力値　divalue　Long (0 or 1)
・DI オン時表示　dionvalue String
・DI オフ時表示　dioffvalue String
・DI 今回入力時刻　ditime　Date
・DI 前回入力値　di1value　Long (0 or 1)
・DI 前回入力時刻　di1time　Date
・オンタイム監視閾値　ontimelimit Long
(単位：秒，監視なしはマイナス値)
・オフタイム監視閾値　offtimelimit Long
(単位：秒，監視なしはマイナス値)
(3) リモート AI　RemoteAI
・AIid　aiid　String
(アプリケーション番号，ノード番号，AI 番号を”-“を介し連接)
・アプリケーション番号　applinumber Long　(0～9)
・ノード番号　nodenumber Long　(0～9)
・AI 番号　ainumber　Long　(0～4)
・アプリケーション名称　appliname　String
(KVS 対応の非正規化のため)
・ノード名称　nodename　String
(KVS 対応の非正規化のため)
・AI 信号名称　ainame　String
・EU 変換パラメータ 1　aitoeupara1 Dounble
（y = a＊x + b の a）
・EU 変換パラメータ 2　aitoeupara2 Dounble
（y = a＊x + b の b）
・EU 単位　eustring　String
（例：Watt，℃，Volt，Ampere）
・AI スキャン周期　aiperiod　Long (単位は秒)
（将来，現在は全 AI はノードごとに一定）
・AI 今回入力値　aivalue　Long (4 桁整数)
・AI 今回 EU 値　aieuvalue　Double
・AI 今回入力時刻　aitime　Date
・Ai 前回入力値　ai1value　Long (4 桁整数)
・AI 前回 EU 値　ai1euvalue Double
・AI 前回入力時刻　ai1time　Date
・AI 前々回入力値　ai2value　Long (4 桁整数)
・AI 前々回 EU 値　ai2euvalue Double
・AI 前々回入力時刻　ai2time　Date
（今回，前回，前々回の値をもつのは，2 次遅れフィルタ $y = c＊x_n + d＊x_{n-1} + (1-c-d)＊x_{n-2}$ の計算のためである．ロギングは今回未実装であるが，これらとは別である．）
・フィルタパラメータ 1 (c)　filterpara1 Dounle
・フィルタパラメータ 2 (d)　filterpara2 Dounle
・AI 上限 2　upperlimit2 Double
・AI 上限 1　upperlimit1 Double
・AI 下限 1　lowerlimit1 Double
・AI 下限 2　lowerlimit2 Double

4.4.4 Java アプリケーション

図 3 の PC（Winodws マシン）において動作するプログラム（試作システムにおけるプログラム名：SerialCommunicationE.java）であり，マイコンへの指示，マイコンからデータ受信，サーバへのデータ送信を行う．今回の試作において不明点も多く，最も苦労した部分である．

(1) マイコンと PC との通信

Arduino の IDE には，「シリアルモニタ」という機能が用意されていて，マイコンと PC との間でデータの送受信が可能である．しかし，今回の試作では，マイコンからデータを受け取ったりマイコンにデータを与えるだけでは，試作の目的を達しない．サーバへのデータ送信というアプリケーション処理を組み込むためには，アプリケーションプログラムを作成する必要があった．

そこで，GAE とも Arduino とも最も近い関係にあると思われる Java 言語を用いてプログラムを作成することにし，参考文献[9]に従った．

参考文献[10]によると，Java でのシリアル通信は，当初の方法（javax.comm.*）が Windows OS では使えなくなっており，「RXTX ライブラリ」を使用する必要があるとのことであった．その指示に従って，2 つの dll ファイル rxtxSerial.dll，rxtxParallel.dll をインストールした．

上記の作業により，Arduino マイコンと PC との間でのシリアル通信（RS232C ポートではなく，USB ポートを使用）が可能になった．

(2) PC とサーバ（GAE）との通信

従来，人間系に閉じた情報システムでは，http の form 機能によって，外部から（すなわち人間から）情報システムにデータを投入していた．今回の試作においても，この延長で考えることにした．すなわち，人間が form でデータ入力する代わりに，マイコンにその動作を行わせることとした．

ネットで検索したところ，見つかったサイトが参考文献[11]であった．それは Desktop.browse による方法で，図 7 のように，GET によるデータの受け渡しを模して PC からサーバのサーブレットを起動することにより，サーバへのデータ送信が可能となった．

```
desktop.browse(new
URI("http://tinyscada.appspot.com/scada?diid="+para
1+"&divalue="+para2));
```

図 7 Desltop.browse による方法（一部）

ただし, この方法によると, 呼び出し側の PC(すなわち, 図 3 における PC にブラウザの残骸が残ってしまうという問題があった．この問題は，サーバにクラウドを用いたことによるものではなく，Desktop.browse の本来機能は，自己の PC にブラウズしたページを表示するという機能であるので，これは当然のことであった．何とか消去する方法がないかネットを検索したが見つからず，1 ヶ月ほど行き詰まりになっていた．

そんな折， 知人から, Http クライアントの方法があることを教示され，ネットで検索した．その結果，参考文献[12]が得られた．これに従って，図 8 のようにプログラムを作成した．

```
strurl="http://tinyscada.appspot.com/scadadi?diid="+p
ara1+"&divalue="+para2;
URL url = new URL(strurl);
HttpURLConnection urlconn =
(HttpURLConnection)url.openConnection();
        urlconn.setRequestMethod("GET");
        urlconn.setInstanceFollowRedirects(false);

urlconn.setRequestProperty("Accept-Language",
"ja;q=0.7,en;q=0.3");
        urlconn.connect();
        Map headers = urlconn.getHeaderFields();
```

図 8 HttpClient による方法（一部）

これにより，ブラウザの残骸が残るという問題は解決した．さらに，この方法によれば，サーバからの情報も獲得することができるので，将来の制御（ディジタル出力）機能に用いることがわかった．ただし，サーバ側からの任意のタイミングでの制御指示ではなくて，クライアントである PC 側からの周期的な問い合わせによる制御機能となる．

サーバからの任意のタイミングによる直接的な制御は，やはり，遠隔側に IP アドレスをもつ必要があり，IPv6 の普及を待つ必要があると思われる．

すなわち，AI スキャン周期（今回のシステムでは，実質的に 10 秒程度以上）程度の時間遅れが許容できるアプリケーション（例えば室温制御など）では使うことができると思われる．

本試作システムを Google にデプロイしている[13]. また, 純正最新のマイコンボード ArduinoUno[14]においても, Japanino 用プログラムがそのまま動作することを確認した.

5. まとめ

今回の試作によって, センサデータをマイコンで収集し, それをクラウドコンピューティング形態にあるサーバに送信し，そこに蓄積するという情報システムの実現性を実験し確認することができた．また，4.4.2 で述べたように，制御機能の実現性も見えてきた．

次の課題としては，この試作システムにデータロギング機能，図 6 の破線範囲外の事象記録機能やグラフ機能を充実させて完成度を高めたい．また，アプリケーションごとにログインで登録ユーザのみのアクセス制御も行いたい．

それと並行して，タイプ B やタイプ C にも取り組んでいきたい．

さらには，実ユーザから実際のニーズがあれば，それにも対応していきたい．

今回は，クラウドによる遠方監視制御の実現性を重点に試作したが 4 月の投稿時点からの進展として，現時点(7 月)では，タイプ B の実現，制御（ディジタル出力）の実現，セキュリティ機能の実現（メンバーログイン機能により，自分のノードのみ監視制御でき，他人のノードにはアクセスできない）, 高機能センサ（センサ側で ADC を行うもの）への対応は達成している．またマイコンボードをケースに入れ製品化へ向けて進展もした．

現時点（7 月）では，ロガー機能，DI 状態変化時のメール発信機能，安全防護（ケーブル短絡によるノードの故障を防止したり，過剰高温時の自動／手動停止）対策，タイプ C の開発などが課題である．

謝辞

本文中にも触れたが，Http クライアントについてご教示いただいた㈲カラビナシステムズの金指文明氏に感謝申し上げる．

また本稿の査読者から有益なコメントをいただき感謝申し上げるが，その反映については十分ではなく，別の機会とさせていただいた．

参考文献

[1] 宮西洋太郎, “クラウドコンピュータとどう向き合うか～クラウドはビジネス系情報システム構築技術の革命となりうるか～”, 信学技報, Vol.110, No.70 pp.1-6 (2010).
[2] 宮西洋太郎, “クラウドコンピューティングトライアルプロジェクトの概要～ソフトウェア産業の繁栄のために～”, 信学技報, Vol.110, No.70 pp.51-56 (2010).
[3] 宮西洋太郎, “クラウドコンピューティングによるアプリケーション試作開発事例～トライアルプロジェクト中間報告～”, 電子情報通信学会 CEATEC2010 連携企画報告，pp.23-30 (2010).
[4] 宮西洋太郎, 廣瀬修, 坂下善彦, 梶功夫, 須栗裕樹, 片岡信弘, “2010 年度クラウドコンピューティングトライアルプロジェクトの総括”, 信学技報, Vol.110, No.427 pp.51-62 (2011).
[5] CCTPにおけるアプリケーションの提案
https://spreadsheets.google.com/ccc?key=0AjPcRg8i-yvcdFZ0Um9NTHdqa05fLXotMmkzb2tKVmc&hl=en#gid=0
ただし登録メンバーのみ.
[6] 放射線災害時にヘリコプターから落下させ, 線量マップを作成する，無線放射線ＧＰＳモニタリングシステム
http://www.ymatic.co.jp/
[7] 大人の科学 Vol.27（2010/5/14）
http://otonanokagaku.net/magazine/vol27/index.html
[8] Arduino http://www.arduino.cc/
[9] Java シリアル通信
http://www.sofix.co.jp/weblog/archives/cat8/java/index.html
[10] RXTX ライブラリ
http://www.maroon.dti.ne.jp/koten-kairo/works/dsPIC/jc2.html
ダウンロードサイト
http://rxtx.qbang.org/wiki/index.php/Download
[11] Desktop.browse による方法
http://javatips.blog62.fc2.com/blog-entry-33.html
[12] Http クライアントによる方法
http://x68000.q-e-d.net/~68user/net/java-http-url-connection-1.html
[13] GoogleAppEngine へのデプロイ
http://tinyscada.appspot/com
[14] ArduninoUno 日本での調達
http://www.switch-science.com/products/detail.php?product_id=428&gclid=CLCFw4H2qagCFVCBpAod4EIhHw