

# Arduino およびクラウド統合型 センサネットワークシステムの開発

森 慎太郎

## 要 旨

近年、防災・災害対策、防災・災害対策、防犯・セキュリティ、医療・介護、交通・流通、施設管理などの様々な分野に対し、無線センサネットワーク技術を応用することが期待されている。しかし、多くの分野での展開が期待されているにも関わらず、現実的に魅力的な活用がなされていない。そこで、本論文では、無線センサネットワークを応用領域において必要とされる無線センサネットワークプラットフォームを開発する。具体的には、特別な知識や技術がなくても無線センサネットワークを実装可能であることを示す。また、大量のセンシングデータを解析・蓄積することが可能なクラウド型システムの構築を図る。本研究の成果により、センシングデータを解析して得られる知見に基づき、関連研究における応用領域での利活用を見出すことに寄与できると考える。

## Abstract

Recently, introducing wireless sensor network into various fields, e.g., disaster prevention, security, medical care, traffic prediction, commodity distribution and facilities management system, has been widely interested. Although wireless sensor network are expected to apply with many fields, practical and attractive services have not been realized and provided. To overcome these situations, we develop a novel wireless sensor network platform. Specifically, our platform can implement the sensor network system without special knowledge and technique. In addition, for analyzing and storing a large quantity of sensing data, we construct the cloud-based network system. We believe that our paper could usefully and effectively help to evaluate the future related studies for the usage and deployment of other application areas.

## 1. はじめに

実世界の情報を収集して我々の社会生活へ還元するための手段として、センシングと無線ネットワークを融合させた無線センサネットワークへの期待が高まっている。無線センサネットワークの利用分野としては、防災・災害対策、防犯・セキュリティ、

医療・介護、交通・流通、施設管理など多岐にわたる。例えば、ホームセキュリティ、ヘルスケア支援のような小規模で身近なものから、橋、道路、森林、農地、都市のモニタリングのような大規模に敷設するものに至っている。他方、これまで無線センサネットワークの技術的な課題とされていた、センサノードの高機能化、自律的なネットワーク構築（センサノードを置くだけでつながる）、電源の確保・省電力化・長寿命化、セルラネットワークを介したインターネットへの接続性に関しては徐々に克服され、すでに利用場所や用途を制限する原因にはなっていない。しかし、多くの分野での展開が

---

A Development of Arduino-based and Cloud-integrated Wireless Sensor Network

Shintaro Mori

福岡大学

Fukuoka University

[解説]

2015年5月18日受付

© 情報システム学会

期待されているにも関わらず、現実的に魅力的な活用がなされていない。

この問題に対処するために、無線センサネットワークの利活用に関する研究がなされている。例えば、文献<sup>[1]</sup>では農業分野において生産性を高める利用手法、文献<sup>[2]</sup>ではヘルスケア分野において健康増進を支援する利用手法、文献<sup>[3][4]</sup>では、スマートビルディング（高度な建物管理）分野において、室内の環境モニタリングおよび適応的な照明・空調の制御手法が提案されている。また、文献<sup>[5]</sup>では、Ambient Assisted Living システムの分野において、居住空間における知的環境モニタリングを統合的に扱う手法、およびモニタリング情報を用いたアクチュエータや家庭ロボットの適応制御、および情報の配信手法が提案されている。

一方、多くの研究者にとって、無線センサネットワークのハードウェアやソフトウェアを実装することには興味がなく、取得したセンシングデータを解析して得られる知見こそが重要である。すなわち、その知見に基づき、応用領域における無線センサネットワークの活用事例を見出すことに意義があるはずである。従って、特別なハードウェアやソフトウェアの知識や技術がなくても、誰でも簡単に無線センサネットワークを開発・構築できる手段、およびセンサノードが時々刻々と生成するような連続かつ大量のセンシングデータをリアルタイムに収集、分析、蓄積する手段があれば、無線センサネットワークの応用研究の発展に寄与できると考える。

以上を鑑みて、本論文では、先述した 2 点の要求を満たすようなオープンスタンダードな無線センサネットワークのプラットフ

ォームを開発することを目的とする。すなわち、前者に関して、センサノードの実装には、オープンソースハードウェアとして開発されている代表的なマイクロコンピュータ Arduino<sup>[6]</sup>を用いる。また、後者に関して、センシングデータを取り扱うためには、サーバ・クライアント型のシステムを用いたのではスケーラビリティに欠けるため、クラウド型システムとして実装する。具体的には、ネットワークシステムの実装には、オープンクラウドとして提供されている代表的なサービスである Amazon Web Services (AWS)<sup>[7]</sup>を用いる。AWS においては、情報システムの稼働に必要な機材や回線はインターネットを介して、必要十分な計算機リソースとしてオンデマンドで利用することができるため、スケーラビリティを担保する無線センサネットワークプラットフォームが構築できる。

無線センサネットワークに関するプラットフォームの開発に関する既存研究として、文献<sup>[3]</sup>では酸性雨の調査支援、文献<sup>[4]</sup>ではヘルスケア支援、文献<sup>[5]</sup>では環境モニタリングを実現するためのプラットフォームが提案されている。センサノードの実装に関して、文献<sup>[3][4]</sup>では特注された専用の端末を用いて構築、文献<sup>[5]</sup>は汎用的に再利用可能な基板を用いて構築している。また、センシングデータを処理するシステムに関しては、文献<sup>[8][9][10]</sup>ともに、アプリケーションサーバを用いて、サーバ・クライアント型のシステムを構築している。そのため、既存プラットフォームは、本論文が目指しているオープンスタンダードな無線センサネットワークプラットフォームの実現には至っておらず、またセンシングデータの解析システム

にスケーラビリティが不十分であるため改善の余地がある。

## 2. 開発した無線センサネットワークプラットフォーム

### 2.1. 開発プラットフォームの概要

図 1 に開発した無線センサネットワークプラットフォームのネットワーク構成を示す。開発したプラットフォームではクラウド型の構成を用いて設計しているため、文献[8][9][10]において用いられているサーバ・クライアント型の構成と比較して、計算機リソースを柔軟に割り当てる事ができる。そのため、スケーラビリティのあるシステム設計を実現することができる。また、特別なハードウェア・ソフトウェアの知識や技術がなくてもセンサノードの実装、およびオープンスタンダードなクラウドサーバ構築を実現するために、Arduino と AWS を組み合わせて開発している。

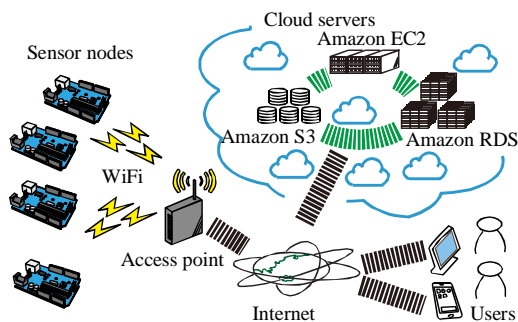


図 1 ネットワーク構成

図 1 において、センサノードで観測したセンシングデータは、アクセスポイントまで無線伝送され (Wi-Fi 方式)、インターネットを介してクラウドサーバに集約される。クラウドサーバの実装には、AWS の中でも Amazon EC2 (Elastic Compute Cloud),

Amazon S3 (Simple Storage Service), Amazon RDS (Relational Database Service) と呼ばれる 3 つの IaaS (Infrastructure as a Service) を用いて構築する。具体的には、Amazon RDS サーバを用いてセンシングデータをデータベースに蓄積し、Amazon EC2 サーバは Amazon RDS サーバとセンサノードまたはユーザの間のインタフェースを担う。すなわち、Amazon EC2 サーバは、センサノードから送られるセンシングデータを受け取り Amazon RDS サーバに記録する。

また、ユーザがセンシングデータを取得できるようにするために、Amazon EC2 サーバは、ユーザからの要求に応じて Amazon RDS サーバからデータを取り出し、ユーザ所望の形式にてデータを開示する働きも併せ持つ。そのため、ユーザはパソコン、スマートホン、タブレット端末を用いてセンシングデータを閲覧することができる。一方、Amazon EC2 サーバおよび Amazon RDS サーバにおける設定データやバックアップデータは、Amazon S3 サーバに保存する。

### 2.2. センサノードのハードウェア実装

図 2 にセンサノードのハードウェア構成を示す。実装したセンサノードは、センサ部、マイクロコンピュータ部、無線通信部、液晶表示部の 4 つのブロックで構成されている。センサ部において、本論文では温度センサと照度センサを実装したが、他の環境モニタリングセンサに置き換えられるため、センサノードを汎用的に利用することが可能である。センサを制御するために、マイクロコンピュータ部には Arduino UNO を用

いる。Arduino UNO には 6 系統のアナログ入出力ピン（10 bit のアナログ・デジタル変換器が標準搭載）が具備されているため、センサの出力ケーブルを Arduino のアナログ入出力ピンに接続するだけで簡単にセンシングデータを取得することができる。

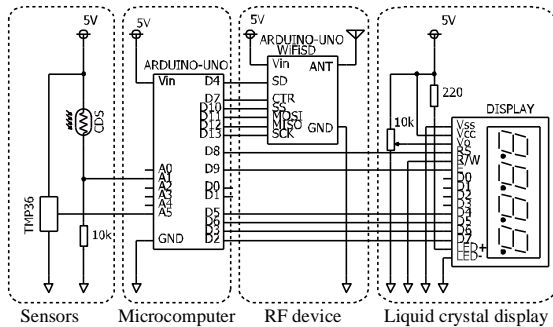


図 2 センサノードのハードウェア構成

センシングデータを無線伝送するために、センサノードとアクセスポイント間に無線通信リンクを確立する必要がある。Arduino においては、機能拡張ボードが数多く提供されており、Arduino UNO に標準 Wi-Fi シールドを組み合わせる（Arduino UNO と標準 Wi-Fi シールドを重ねて組み立てる）ことによって、無線通信部を実現できる。液晶表示部に関しては、センサノードを試作する際のデバッグ情報を表示するために実装している。液晶表示部の実装は、液晶パネルキットの入出力ピンと Arduino UNO の汎用入出力ピンを接続することにより実現できる。

### 2.3. センサノードのソフトウェア実装

図 3 に実装したセンサノードにおけるセンシング処理の信号処理手順を示す。無線通信部の初期化処理の後、センシング処理を行うトリガーを生成するためにタイムスロット判定を行う。一般にトリガー生成は

タイマー割り込みを用いるべきであるが、Arduino には標準で実装されていない。そこで、一定周期でカウントされているカウンタを用いて、所望の周期でトリガーを発生させることで代用する。無線伝送リンクの確立を確認（必要に応じて無線通信リンクの確立処理を実行）し、アナログ入出力ピンに接続されているセンサの電圧を計測してセンシングデータを生成する。センシングデータは、オープンデータとして汎的に利用可能な Extensible Markup Language (XML) 形式にて定型化する。そして、クラウドサーバに Hyper Text Transfer Protocol (HTTP) の POST リクエストにてクラウドサーバに投稿する。

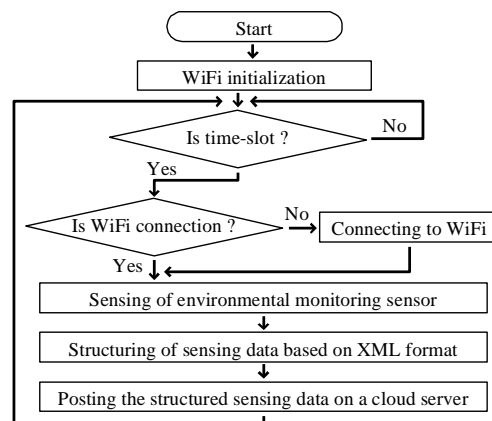


図 3 センサノードの信号処理手順

図 4 に、図 3 の信号処理を実現するためのプログラムコードを示す。Arduino へのコーディングには、C/C++言語に類似した専用のプログラミング言語を用いて実装する。具体的には、プログラムコードは、大別して setup メソッド、および loop メソッドと呼ばれる 2 つのメソッドから構成されている。setup メソッドには、初期化処理のような Arduino が起動した際に一度だけ実行される処理を記述し、loop メソッドは、

setup メソッドが実行された後に繰り返し実行される処理を記述する。また、あらかじめ Arduino の標準ライブラリに規定されている組み込み型メソッドのほかに、ユーザ独自の処理を記述したユーザ定義型メソッドを追加することもできる。表 1 において、図 4 に示すプログラムコードで使われているメソッドの仕様をまとめた。

```

01: #include <SPL.h>
02: #include <WiFi.h>
03:
04: void setup()
05: {
06:   // Initializing the serial port for debug usage.
07:   init_serial();
08:   // Initializing the WiFi connection.
09:   init_wifi();
10: }
11:
12: void loop()
13: {
14:   // We transmit the sensing data to the cloud server every time-slot.
15:   if ( millis() - last > interval )
16:   {
17:     // If WiFi connection is disable, we establish a new connection.
18:     if (!client.connected())
19:       connect_wifi();
20:
21:     // We obtain the node number, temperature and illuminance values.
22:     const int node = 1;
23:     double temp = get_temperature();
24:     int illu = get_illumiance();
25:
26:     // We structuralize the sensing data based on XML format.
27:     String xml_data =
28:       xml_format ( node, temp, illu );
29:
30:     // We submit the structured sensing data with HTTP/POST.
31:     http_request ( xml_data );
32:   }
33: }

```

図 4 Arduino に実装するプログラムコード

表 1 Arduino に実装するプログラムコード内のメソッドの仕様

名称	種別	機能
init_serial	ユーザ定義型	デバッグ用の通信リンクを確立する
init_wifi	ユーザ定義型	Wi-Fi リンクの初期化処理を行う
Milis	組み込み型	プログラムが実行後の経過時刻を返却する
connect_wifi	ユーザ定義型	Wi-Fi リンクへの再接続処理を行う
get_temperature	ユーザ定義型	現在の温度を計測する
get_illumiance	ユーザ定義型	現在の照度を計測する
xml_format	ユーザ定義型	XML 形式に定型化する
http_request	ユーザ定義型	HTTP リクエストにてクラウドサーバに投稿する

## 2.4. クラウドサーバのシステム構築

図 5 にセンシングデータをクラウドサーバに登録する処理に関して、センサノードとクラウドサーバ間の信号処理手順を示す。はじめに、センサノードはアクセスポイントとの間に無線通信リンクを確立して、Amazon EC2 サーバとの間に Transmission Control Protocol (TCP) セッションを構築する。次に、センサノードは、XML 形式に定型化したセンシングデータを Amazon EC2 サーバの受付処理を担う Common Gateway Interface (CGI) インスタンスに HTTP の POST リクエストで投稿する。CGI インスタンスは、XML 形式のセンシングデータを構文解析した後、Amazon RDS サーバ上の SQL データベースにセンシングデータを登録する。CGI インスタンスは、図 6 (a) に示すように、Perl 言語を用いて実装した。すべての処理が完了した後、センサノードとクラウドサーバ間に確立された TCP セッションおよび無線通信リンクを切断する。

一方、ユーザからの要求に応じてセンシングデータを開示する処理を実現するために、Amazon EC2 上に、図 6 (b) に示す CGI インスタンスを図 6 (a) とは別に実装した。具体的には、CGI インスタンスは、SQL サーバからセンシングデータを読み出し、ユーザ所望のデータ形式に定型化した後、HTTP の GET リクエストにて応答する。

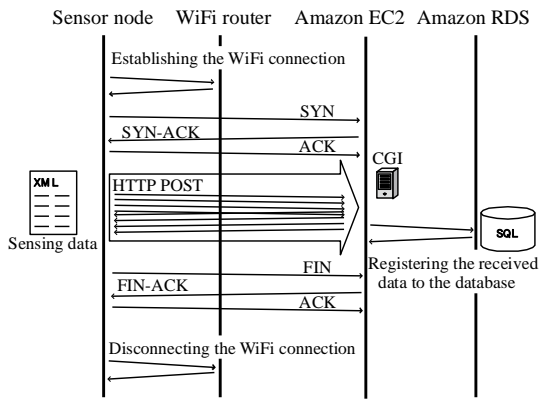


図5 センサノードとクラウドサーバ間の信号処理手順

```
01: #!/usr/bin/perl
02:
03: # We obtain the sensing data by using HTTP/POST method.
04: read(STDIN, $post_data, $ENV{'CONTENT_LENGTH'});
05:
06: # We decode and parse the XML-based structured sensing data.
07: use XML::Simple;
08: $parser = XML::Simple->new;
09: $xml = $parser->XMLin($post_data);
10: $node = $xml->{head}->{node_number};
11: $temp = $xml->{body}->{temperature};
12: $illu = $xml->{body}->{illuminance};
13:
14: # We register the unstructured sensing data to the SQL database
15: use DBI;
16: $dbh = DBI->connect('DBI:mysql:...', 'taro', 'password');
17: $stmt = $dbh->prepare("INSERT INTO sensing_data
18:   VALUES($node, $temp, $illu);");
19: $stmt->execute;
20: $stmt->finish;
21: $dbh->disconnect;
22:
23: # We display the results of processing
24: print "Content-type: text/html \n\n";
25: print "OK\n";
```

(a) Acceptance and database registration transactions

```
01: #!/usr/bin/perl
02:
03: # We obtain the sensing data from the SQL database.
04: use DBI;
05: $dbh = DBI->connect('DBI:mysql:...', 'taro', 'password');
06: $stmt = $dbh->prepare("SELECT * FROM sensing_data;");
07: $stmt->execute;
08: for (; $ref = $stmt->fetchrow_arrayref)
09: {
10:   @buffer = @$ref;
11:   push (@result, \@buffer);
12: }
13: $stmt->finish;
14: $dbh->disconnect;
15:
16: # We display the sensing data according to the HTML format.
17: print "Content-type: text/html \n\n";
18: &print_html_view (\@result);
```

(b) Exhibition transactions of sensing data

図6 Amazon EC2 に実装した受付処理プログラムコード

### 3. 評価

#### 3.1. 動作検証

図7 にセンサノードの動作検証の様子を

示す。実装したセンサノードは、Arduino UNO, 標準 Wi-Fi シールド, 温度センサと照度センサを実装した試作シールドを3段に重ねて構築した。センサノードのデバッグおよび電源供給のために、USB ケーブルを介してパソコンと接続した。当然、電池やバッテリーを用いた単独動作も可能であり、センシングデータは USB ケーブルを介して通信していない。図7の右上側に、センサノードの状態を表示したシリアルモニタの画面を拡大して掲載した。シリアルモニタの表示より、センサノードにおいて、Wi-Fi の初期化処理が正常に行われ、一定時間ごとにクラウドサーバにセンシングデータを投稿できていることを確認した。

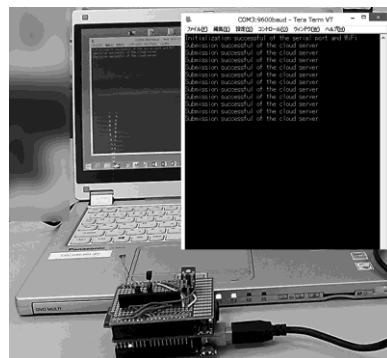


図7 センサノードの動作検証

ユーザがセンシングデータをパソコンおよびタブレット端末を用いて閲覧することを想定して、クラウドサーバからセンシングデータを取得する検証を行った。図8にパソコン、図9にタブレット端末での閲覧結果を示す。本検証では、Hyper-Text Markup Language (HTML) 形式にてクラウドサーバからセンシングデータを取得した。HTML 形式によるデータの取得が可能であるため、Amazon EC2 サーバの CGI インスタンスを書き換えることにより、XML

形式に基づく種々のアプリケーションサービスへの応用も可能である。例えば、Resource Description Framework (RDF) サービス, Really Simple Syndication (RSS) サービス, Java Script Object Notation (JSON) サービスなどが挙げられる。

イベントID	時刻	センサーID	温度センサ値	湿度センサ値
1	2014-09-18 10:37:54	1	25.19531	86
2	2014-09-18 10:38:57	1	24.70703	86
3	2014-09-18 10:40:02	1	24.70703	87
4	2014-09-18 10:41:03	1	25.19531	87
5	2014-09-18 10:48:24	1	26.17188	87
6	2014-09-18 10:49:27	1	25.19531	87
7	2014-09-18 10:50:31	1	25.68359	86
8	2014-09-18 10:51:34	1	25.68359	86
9	2014-09-18 10:57:35	1	23.73047	84
10	2014-09-18 10:59:08	1	22.75391	82
11	2014-09-18 11:01:52	1	24.21875	83
12	2014-09-18 11:02:55	1	22.75391	82
13	2014-09-18 11:04:54	1	23.73047	83
14	2014-09-18 11:05:57	1	22.75391	83
15	2014-09-18 11:07:00	1	23.24219	83
16	2014-09-18 11:08:03	1	22.75391	82
17	2014-09-18 11:09:06	1	22.75391	83
18	2014-09-18 11:10:09	1	22.75391	83
19	2014-09-18 11:11:12	1	23.24219	80
20	2014-09-18 11:12:15	1	23.24219	82

図8 センシングデータをパソコンから閲覧

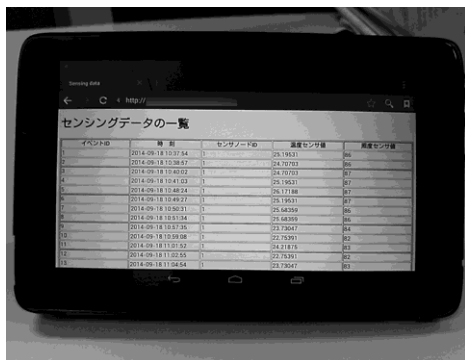


図9 センシングデータをタブレット端末から閲覧

### 3.2. 考察

3.1 節において、開発した無線センサネットワークプラットフォームの動作検証を行った。本節では、開発したプラットフォームに対して、ソフトウェアの知識や技術がなくても、誰でも簡単に無線センサネットワークを開発・構築できる点、およびセンサノードが時々刻々と生成するような連続かつ

大量のセンシングデータをリアルタイムに収集、分析、蓄積できる点が満たされているか考察する。

前者に関して、センサノードは Arduino を用いて実装しているため、マイクロコンピュータに基づくシステム開発は簡単に実現できる。この点に関して、工学部(情報系)の学部3年生が初めてマイコン開発を行い、個人差はあるが、2~3カ月で所望のセンサノードを実装することができている。とくに、Arduino に具備されているアナログ入出力ピンにセンサを取り付けるだけでセンシング可能である点、無線伝送処理を実現する IC チップ群が標準 Wi-Fi シールドとして提供されているため、メイン基板と組み合わせるだけで実現できる点が実装容易性を高めている。また、ソフトウェアの実装に関しては、C/C++言語に類似したプログラミング言語が採用されているため、その習得も容易である。

後者に関して、サーバ・クライアント型システムとは異なり、クラウド型システムを採用しているため、無線センサネットワークシステムをスケーラブルに実装できている。とくに、センサネットワークが収集するセンシングデータは、ビッグデータと呼ばれる膨大な演算処理およびデータ蓄積が必要である。この点に関して、柔軟に計算機リソースを割り当てることが可能であるため要件を満たしている。また、AWS の場合、オープンクラウドと呼ばれるクラウド型システムであるため、センシングデータの処理だけではなく、多数のユーザへセンシングデータを提供する際の負荷分散処理についても、インターネットを介して計算機リソースをスケーラブルに割り当てることが

できる。

以上のことから、開発した無線センサネットワークプラットフォームは、所望の要件を満たしている。

#### 4. おわりに

本論文では、オープンスタンダードな無線センサネットワークプラットフォームを実現するために、Arduino および AWS を用いた新たなプラットフォームを開発した。今後の課題は、本プラットフォームを用いて実環境における利活用での評価、センサノードのソフトウェア実装を支援するためのツール開発がある。また、センシングデータを解析して得られた知見に基づき、新たな応用領域を創造する必要がある。

#### 参考文献

- [1] Y.Shouyi, L.Leibo, Z.Renyan, S.Zhongfu, and W.Shaojun, "Design of wireless multi-media sensor network for precision agriculture," *IEEE Commun. China*, Vol.10, No.2, pp.71-88, Feb. 2013.
- [2] A.Solanas, C.Patsakis, M.Conti, I.S.Vlachos, V.Ramos, F.Falcone, O.Postolache, P.A.Perez-Martines, R.D.Pietro, D.N.Perrea, and A.Martinez-Balleste, "Smart health: A context-aware health paradigm within smart cities," *IEEE Commun. Mag.*, Vol.52, No.8, pp.74-81, Aug. 2014.
- [3] T.Torfs, T.Sterken, S.Brebels, J.Santana, R.Hoven, V.Spiering, N.Bertsch, D.Trapani, and D.Zonta, "Low power wireless sensor network for building monitoring," *IEEE Sensors Journal*, Vol.13, No.3, pp.909-915, Mar. 2013.
- [4] N.K.Suryadevara, S.C.Mukhopadhyay, S.D.T.Kelly, and S.P.S.Gill, "WSN-based smart sensors and actuator for power management in intelligent buildings," *IEEE/ASME Trans. Mechatronics*, Vol.20, No.2, pp.564-571, Apr. 2015.
- [5] J.Lloret, A.Canovas, S.Sendra, and L.Parra, "A smart communication architecture for ambient assisted living," *IEEE Commun. Mag.*, Vol.53, No.1, pp.26-33, Jan. 2015.
- [6] Arduino, <http://www.arduino.cc/>.
- [7] Amazon web services: <http://aws.amazon.com/>.
- [8] B.Zhou, S.Yang, T.H.Nguyen, T.Sun and K.T.V.Grattan, "Wireless sensor network platform for intrinsic optical fiber pH sensors," *IEEE Sensor J.* Vol.14, No.4, pp.1313-1320, Apr. 2014.
- [9] S.Junnila, H.Kailanto, J.Merilahti, A.-M.Vainio, A.Vehkaoja, M.Zakrzewski and J.Hyttinen, "Wireless multipurpose in-home health monitoring platform: Two case trials," *IEEE Trans. Info. Tech. in Biomedicine*, Vol.14, No.2, pp.447-455, Mar. 2010.
- [10] M.T.Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE J. Emerging and Selected Topics in Circuits and Systems*, Vol.3, No.1, pp.45-54, Mar. 2013.