

Arduino, ZigBee, OSC を活用した小規模情報共有環境の構築と 非常時情報通信への応用

大野 浩 之^{†1}

非常時における情報通信のあり方を研究対象としている著者にとってとても興味深く、かつ急速に普及しつつある技術や製品に "ZigBee", "OSC (Open Sound Control)", "Arduino" がある。本報告では、これらを適切に組み合わせたプロトタイピングを ProtoZOA アプローチと呼び、この方法を用いることで、大学の研究室等の小規模な組織における情報共有環境がこれまでとは異なる形態で構築可能になることを指摘する。本報告では、すでに製作したいいくつかの実装について述べ、そのうちのいくつかは非常時情報通信への応用が可能であることを示す。

Design and Implementation of an Information Sharing Environment using Arduino, ZigBee and OSC, And Its Application to Emergency Communication Systems

HIROYUKI OHNO^{†1}

Some of new technologies and new products which are recently available widely with a low price are very interesting for us who are studying the emergency communication systems. They are ZigBee, "OSC" (Open Sound Control), "Arduino". We call the new approach "ProtoZOA", which used them. Using ProtoZOA, By using ProtoZOA Approach, construction of the information-sharing environment in a small-scale organization is attained with a different form from the former based on WiFi and TCP/IP. In this report, many implementations based on the ZOA approach are reported, and we noticed some of them are available for the emergency communications.

^{†1} 金沢大学
Kanazawa University

1. はじめに

非常時における情報通信のあり方を研究対象としている著者から見て研究遂行上興味深い技術や製品が、この 1,2 年の間にこれまでに安価かつ容易に入手できるようになってきた。特に注目しているのが以下に列挙した "Z", "O", "A" の 3 つである。

- (1) Z (ZigBee)¹⁾ – アドホック無線通信環境を手軽に実現する短距離無線通信方式。ZigBee に対応した小型モジュールが普及しつつある (その頭文字から以後 "Z" で表す)。
- (2) O (OSC (Open Sound Control))²⁾ – URI に似た形式の独自の名前空間に値を割り当て、メッセージとして送付する手法。さまざまなプラットフォームに OSC ライブラリが移植されており利用しやすい (その頭文字から以後 "O" で表す)。
- (3) A (頭文字が A のいくつかの技術や製品) – ARM 社がライセンスする ARM アーキテクチャの CPU や ATmel 社の AVR マイクロコントローラのビジネス的な成功と、AVR を採用した "Arduino"³⁾ のような「新世代の組み込み型ハードウェアとその開発環境」の急速な普及。これらとは切り口は異なるが、Apple 社の製品群 (iPod Touch, iPhone, iPad) や、Google Android も興味深い (偶然ではあるが、これらの頭文字はみな A なので、以後これらを "A" 表す)。

上記の "Z", "O", "A" の詳細については次節以降で述べるが、これらを適切に組み合わせることで、研究室の情報共有環境が改善され研究活動がこれまでより効率化し、研究上の新たな情報共有に係る取り組みも実施しやすくなると判断した。そこで、これらを組み合わせた情報共有環境と、この環境を活用したプロトタイピングの推進に今後積極的に取り組むこととし、「"Z" "O" および "A" を活用した試作環境 (Prototyping by "Z", "O" and "A")」の意味で「ProtoZOA 環境 (あるいは ProtoZOA アプローチ)」と呼ぶことにした。本報告では、2010 年 4 月から開発を始めた ProtoZOA 環境 (ProtoZOA アプローチ) の現状と今後の展開について述べる。

2. 本研究の目的

本研究では、"Z", "O", "A" の 3 要素を組み合わせることで何が実現でき何が改善されるかを事前に検討した上で実装して評価することになる。評価の軸は少なくとも二つあり、ひとつは、著者らの非常時情報通信環境の研究開発に直接寄与するか否かで、もうひとつは、非常時情報通信環境の構築を行う著者らの研究開発活動の改善に寄与するかである。前者に沿った成果は、非常時情報通信環境を研究テーマにする者にしか寄与しないが、後者は

情報通信分野の研究室の環境改善であると解釈できるので、多くの研究開発者の参考になるはずである。

3. ProtoZOA の構成要素と "Z" "O" "A" の連携がもたらす効果

本節では、本研究における "Z", "O", "A" の意味を述べ、これらの連携がもたらす効果を指摘する。

3.1 "Z" – ZigBee

"Z" は ZigBee である。ZigBee は、短距離での利用を主な用途とする無線通信方式のひとつであり、その下位層（おおむね OSI 第 2 層以下）は、IEEE802.15.4 として標準化されており、それより上位は、ZigBee アライアンスという団体が仕様を策定している。日本国内では 2.4GHz 帯を用い、物理層の最大通信速度は 250kbps であるが実効速度はその 1/5 ~ 1/10 であるとされている。しばしば、赤外線リモコンの代替技術と称され室内での利用のみが注目されるが、適切なアンテナをつけて屋外の見通しの効く環境で運用することで、最大で 1.6km 程度の通信が可能な製品もある。消費電力を削減するため、10 ミリ秒台の時間でスリープあるいはサスペンドが可能になっている。この機能によりたとえば 1 分に一度何かを計測して伝送するアプリケーションでは、1 分のうち 59.9 秒間はスリープし、0.1 秒間だけ起動してその間にデータを採取して転送し再びスリープするといった対応が可能である。ZigBee ではメッシュ型ネットワークが実現可能で、二つのノードが複数の中継ノードを介して通信できる。その際、ユーザはそのことを意識する必要はない。ZigBee はセンサネットワークに投入することを念頭においているが、データ量が少なればより一般的なデータ通信に利用可能である。また、IEEE802.15.4 上で IPv6 を動作させる "IPv6 over Low-Power Wireless Personal Area Networks" (6LowPAN) も議論されており、今後は低速は承知の上で IPv6 化が進む可能性が高い。

3.2 "O" – OSC

"O" は、OSC (Open Sound Control) である。OSC は、電子楽器同士や電子楽器とコンピュータを接続することを念頭にカリフォルニア大学バークレー校で開発された通信プロトコルで、1980 年代に同様の主旨で開発された MIDI の後継とされている。OSC は、URI のようなスラッシュで区切った名前空間を持ち、それぞれの名前にデータ型と値を対応させてメッセージとする。このメッセージを本報告では OSC メッセージと呼ぶ。OSC においては、OSC メッセージをブロードキャストによって共有することが多く、IP 環境においては UDP/IP を用いてブロードキャストすることが多いが、これは必須要件ではないので、

ProtoZOA アプローチでは ZigBee ネットワーク上で OSC メッセージをユニキャストでやりとりしている。

OSC は、アプリケーション層（あるいはそれに近い）プロトコルであり、経路制御やリンクの確立には関与しない。このため OSC の実装は比較的容易であり、すでにさまざまなプラットフォームのさまざまな言語やアプリケーションに実装されている。この点は、ProtoZOA 環境をさまざまな言語やアプリケーションから利用可能にする際に大きな利点となる。

3.3 "A" – Arduino

上述のように、"A" には、偶然ではあるがさまざまな素子や機器が該当する。広義にはこれら全てを指すが、狭義には Arduino のみを指す。本報告では特に断りがない限り "A" は、狭義の "A" である。

Arduino は、ATmel 社の AVR マイクロコントローラを用いたさまざまなサイズのマイクロコントローラボード（最大でも名刺サイズ）と、Arduino で動かすバイナリイメージを記述するための C++ 風の言語（しばしば Arduino 言語と称する）と当該言語のためのソフトウェア統合開発環境の総称で、2005 年にイタリアで開発が始まった。Arduino のソフトウェアは GPL v2 で、ハードウェアは Creative Commons Attribution Share-Alike 2.5 ライセンスで提供されている。広義の Arduino は、ハードウェア部分だけでなく言語や開発環境も指すが、狭義には、ハードウェア部分のみを指す。本報告で特に断りなく Arduino と記した場合には狭義の Arduino を指す。Arduino のハードウェアにはいくつかの形状があるが、名刺大のものは（現在流通している機種では Duemilanove や UNO が該当する）同じ大きさの拡張基板（Arduino 用語では「シールド」(shield) という）をピンヘッダを介して積層できるようになっている。Arduino 言語により、ピン単位のデジタル入出力、アナログ入出力や従来型の非同期シリアル通信に加え、I2C, SPI といった、2 線式、3 線式の 1 対多通信にも対応している。

3.4 "Z", "O", "A" による小規模組織での情報共有環境の改善

ProtoZOA では、ZigBee モジュールを搭載した Arduino が ZigBee プロトコルで通信する環境を作り、この環境上で OSC メッセージをやりとりして情報を共有する。

同様のことは、これまでも無線 LAN 機能を搭載したパソコンで IPv4 や IPv6 を使って通信する環境を作り、この環境上で、HTTP や SMTP といった各種アプリケーションプロトコルを使ってメッセージをやりとりして実現してきたので、発想そのものは特段新しいものではない。しかし、ProtoZOA においては、"Z", "O", "A" のそれぞれがこれまでの方法の対応する部分に比べて小さく軽量にできている。たとえば、"A" を構成する Arduino

は、8bit CPU の AVR を用いており、プログラムを動かす際に関与する一次および二次記憶装置のサイズは数十キロバイトのオーダーであって、その小ささは現行のパソコンとは比べものにならない。通信する場合も、インターネット上の相手と直接通信することよりは、同一ネットワーク上の相手と通知することが多く、OSI 3 層における経路制御などは、必要であれば適宜プロキシを用意するなどして対応できる。よって、“A” 上に実装しなくてよい。このような環境では、フルスペックの TCP/IP 環境を用意するのではなく、小型軽量でありながらアドホックネットワークを構築する ZigBee のような通信プロトコルは相応である。なお、“A” の上に “Z” を構築した場合、そこでやりとりされる情報の扱いに統制がないと混乱を招くことも有り得る。そこで、ProtoZOA では、OSC メッセージを導入している。これは、従来の環境でいえば、TCP/IP 環境でメッセージをやりとりする際に、TCP/IP 上で無形式の平文をやりとりするのではなく XML などの形式のあるデータで情報をやりとりすることに相当する。以上より ProtoZOA アプローチは理にかなっている。

ProtoZOA アプローチにも問題はあつた。たとえば、Arduino で処理能力は足りるのか (8bit CPU と小さなメモリ)、ZigBee の通信速度は足りるのか (物理層の速度でさえ最大 250kbps であり、実質的にはその 1/5 ~ 1/10 の 20kbps ~ 50kbps 程度となる)、OSC メッセージで必要十分なのかといった疑問は容易に提起できる。これらを精査することは今後の課題となるが、後述する現時点での実装例が示すように、現時点ではこれらの選択に深刻な問題は生じていない。

ところで、本報告ではここで言及するに留めるが、このような極小の環境にも IPv6 を取り込んだ小型 OS を開発して搭載しようという contiki⁴⁾ プロジェクトのような新たな活動や、1980 年頃に 8bit CPU 向けに商品として提供され後に公開された CP/M を復活させようという動きがある。今後、ProtoZOA アプローチがこれらの成果を取り込むことは否定しない。

3.5 -ARK プロジェクトとの関係

ここで、著者が 2006 年度より推進している -ARK プロジェクト⁵⁾ と、ProtoZOA アプローチの関係について言及しておく。-ARK プロジェクトは、大規模自然災害等の非常時の最も初期にあたる自助および共助が必要な時期に資する情報通信環境を日頃から使っているスマートホンや携帯端末上に構築することを目指している。-ARK プロジェクトは、発足当初はハードウェアの製作は行わない方針だったし、通信は無線 LAN 上で TCP/IP で行うことを暗黙の前提としてきたので、発足当初の方針を堅持する限り、ProtoZOA との接点はあまりない。しかし、ハードウェア製作は行わず、WiFi、TCP/IP に固執する必

要は特になので、今後は ProtoZOA の成果を適宜採り入れる。本報告の後半で言及する -ARK/Radio、-ARK/ALERT、-ARK/L7GW などはずでに -ARK プロジェクトに取り込んでいる。

4. ProtoZOA の設計

4.1 基本方針

基本方針として、“Z”、“O”、“A” とともに既存の情報や機材を極力活用し、新規開発は他に利用可能な資源がない場合に限定することとする。

4.2 Arduino の利用方法

Arduino には、Arduino 本体が名刺大の機種に接続可能な多種多様なシールドが製品あるいはキットの形で流通している。

基本方針に従い、製品あるいはキットがある場合には、新規に設計せずにこれらを利用することにした。本報告で言及する実装例のうち、イーサネット (IPv4)、MIDI、DC モータ制御などはキットを利用している。製品もキットもない場合には、汎用シールド (ユニバーサルシールド、パニラシールドなどということもある) を用いて必要な機能を設計実装することになる。実装例では、-ARK/Radio、-ARK/ALERT、LCD 付無線コントローラ、電源制御アダプタなどが該当する。

ProtoZOA では、“A” に “Z” や “O” を組み込んで実現するが、Arduino だけで全てが完結するわけではなく、既存のインターネットとの世界とのゲートウェイも必要である。そこで、“A” に属さないパソコンのような装置に “Z” や “O” を組み込んで、“ZOA” の世界とこれまでの TCP/IP + LAN の世界とのゲートウェイを作るか、Arduino にイーサネットインタフェースを取り付け、Arduino でゲートウェイを作る必要がある。なお、Arduino 用の WiFi シールドは存在するが、現時点では日本国内では使用できない。

4.3 ZigBee の運用方法

ZigBee を実装したモジュールは何社かから発売されているが、今回は入手の容易さから Digi International 社の XBee モジュールを用いることとした。

ZigBee ネットワークを構成する各ノードは、必ず「コーディネータ」「ゲートウェイ」「エンドデバイス」のどれかになる。このうち「コーディネータ」はひとつの ZigBee ネットワークに 1 台しか存在できないが「ゲートウェイ」「エンドデバイス」は複数台存在できる。XBee モジュールを用いて ZigBee ネットワークを構築するには、最初に目的に応じた最新のファームウェアを書き込み、さらに適切な設定を行う必要がある。設定項目は多岐に及ぶ

が、最低限 PAN ID という 64bit 長の ID を指定しなければならない。ZigBee ネットワークは PAN ID を用いてネットワークを識別しているからである。

もし、ある空間に複数の ZigBee ネットワークが存在し、それらのうちのいずれかに接続したい場合には、PAN ID を指定して ZigBee ネットワークに接続する必要がある (ZigBee ネットワークがひとつしかなければその ZigBee ネットワークに自動接続する方法もある)。ここで、PAN ID をどうやって伝えるかと、複数の ZigBee ネットワークが同じ空間に存在する可能性を考慮して互いに衝突しない PAN ID をどうやって決めるかについて検討しなければならない。

ありがちな方法は、誰かが PAN ID 管理者になり、管理者から発行してもらった PAN ID を使うという方法だが、ProtoZOA アプローチでは、管理者が存在しなくても衝突しない PAN ID を以下のようにして発行することにした。すなわち、コーディネータとなるモジュールの所有者のメールアドレスをもとにそのハッシュを求め、求めたハッシュの下位 64bit を PAN ID として使うという方法である。もし 64bit が長すぎるのであれば、適宜短くして上位はゼロで埋めるものとする。

こうして作成した PAN ID が、他のメールアドレスをもとに作った PAN ID と衝突する確率は、情報セキュリティにおける「誕生日攻撃問題」が示すとおりであり、PAN ID の実質的な bit 長が N bit の場合、およそ $2^{(N/2)}$ 個の異なるネットワークが同時に存在した場合に、衝突確率が 5 割を超える。よって、現在の ZigBee ネットワークの存在頻度なら、32bit 程度でも十分である。

この方法であれば、コーディネータとなるモジュールの所有者のメールアドレスさえわかれば PAN ID は確定し、当該 ZigBee ネットワークに参加できる。逆に、この方法がセキュリティ上好ましくなかったり、同一人物が複数の PAN ID を必要としている場合には、メールアドレスに何らかのキーワードを加えた上でハッシュを求めることとし、「何らかのキーワード」を当該 ZigBee ネットワーク関係者の中で取り決めることにする。この場合も、何らかのキーワードを同一コミュニティ内の人間であれば推測できるようにすれば、部外者だけを排除する方式になるし、その都度伝えるのであれば部外者にも推測できない PAN ID を持つ ZigBee ネットワークを構築できる。

4.4 OSC メッセージの利用方法

OSC メッセージで用いる名前は、URI あるいは UNIX 系 OS のファイルシステムにおけるパスに似た、"/" (スラッシュ) で始まり、"/" を区切りとして名前を連ねて行く方式である。OSC メッセージではそれぞれの名前にデータ型 (32bit 整数型、小数型、バイナリ

型、文字列型等) と対応する値を割り付ける。OSC の一般的な使い方では、OSC メッセージは個人が把握できる範囲を想定しているため、第三者が彼らのために設定するの名前との衝突は考慮しないのが普通である。一方、ProtoZOA アプローチでは、ProtoZOA 対応の装置を他の ProtoZOA 環境に移しても動作するように、OSC メッセージにはある程度のルールを導入しておくのがよい。OSC コミュニティに独自の名前空間の最初の部分を登録できるようだが、そこまでしなくても以下のようにすると衝突はかなり回避できる。なお、OSC メッセージの名前の全長は最大 256 文字である点に配慮する必要がある。

- /プロジェクト名/ユーザ名/...
- /プロジェクト名/ZigBee PAN ID/...

5. ProtoZOA の実装状況

以下に 2011 年 2 月の時点で ProtoZOA アプローチで実装を終えた機材 (一部、改良中の機材を含む) を列挙する。ProtoZOA 環境と ProtoZOA 環境を取り巻く従来の環境との間の情報の流れに着目して 3 つに分類して記しているが、便宜上そうしただけで、厳密な分類ではない。

5.1 グループ A - ProtoZOA から外部へ向けた情報の発信・表現

このグループに属する実装は、ProtoZOA 環境から ProtoZOA 環境の外へ情報を送るものである。

5.1.1 Twitter

Arduino にイーサネットシールドを取り付け、Twitter に向けてメッセージをつぶやく ProtoZOA 実装である。現在 Twitter にメッセージを自動でつぶやくには OAuth に対応する必要があり、Arduino のような処理能力の小さなデバイスの場合は、あらかじめトークンを取得して StewGate 1⁶⁾ というプロキシ経由でつぶやく場合が多い。ProtoZOA 環境の中から Twitter に自動的につぶやけるようになったことで、ProtoZOA を情報発信環境とみなせるようになった。現状では以下の OSC メッセージ (データ型は文字列) で、英数字のメッセージをつぶやける。

```
/e-ARK/hohno-01/Expr/Out/Twitter-01
```

なお、ZigBee でブロードキャストが使えるのであれば、上記の OSC メッセージをブロードキャストすれば問題ないが、何度か試したものの効率がよくなかったという理由で、現状ではブロードキャストは行っておらず、送り側がメッセージを送り届ける受け手側の ZigBee モジュールの MAC アドレスを直接指定してユニキャストしている。この方法に至ったの

は、ブロードキャストメッセージの配信が現状では効率良く行えないからであるが、OSC はメッセージをブロードキャストすることを暗黙の前提としているので、ブロードキャストができないと OSC メッセージを使う魅力が半減してしまう。次報までにはこの問題には対処して報告したい。

5.1.2 電源制御

ネットワーク上の機材から周囲の電源を制御したいというのはしばしば生じる要求である。電源を制御するためにリレースイッチや SSR (ソリッドステートリレー) などが利用されている。Arduino からこれらを制御することは何ら問題なくできるが、現状では以下の OSC メッセージ (データは二つあり、データ型はいずれも整数) で、指定した番号に対応した電源を制御できるが、電源制御は対象を誤ると事故に至るので、見直しを図る予定である。現状では第一引数が制御するリレーや SSR の番号で、第二引数がゼロならオフ、それ以外ならオンである。

```
/e-ARK/hohno-01/Expr/Out/PowerControl-01
```

また、一部の電源制御系には電流センサが組み込んである。すでに、消費電流は A/D 変換した上で読みだしているが、これをどういう形式の OSC メッセージにするかは調整中である。次項で述べるようにセンサ関連全体の名前空間の設定が遅れているためである。

```
/e-ARK/hohno-01/Expr/In/PowerControl-01
```

5.1.3 モータ制御

著者の周囲には、何らかの情報の変化を実際の動きで表現したいという要求がある。たとえば、ネットワークトラフィックの変化を、アナログ式メータの針が動くような動作で表現したいとか、モータの回転数の多少に対応させたいという要求が現実にある。このため、OSC メッセージを受け取った Arduino が、メッセージに対応してサーボモータや DC モータの動きに対応できるようにした。現状では以下の OSC メッセージでモータを制御している。Motor-NN の NN はモータの番号である。現状ではサーボモータと DC モータとステッピングモータをひとまとめにしているため、引数の数や意味は、モータごとに異なる。このあたりの不統一はできるだけ早く整理したい。

```
/e-ARK/hohno-01/Expr/Out/Motor-NN
```

5.1.4 ディスプレイ

流通している Arduino に接続可能な機材には、表示装置が多い。モノクロ液晶、カラー液晶などはさまざまな種類が流通しているし、マルチドット LED や多連装 7 セグメント LED や 16 セグメント LED もある。これらの多くが、非同期シリアル通信、I2C、SPI (主として

液晶) やダイナミックドライブ方式で点灯している。ZigBee シールドを搭載した Arduino にこれらの表示器を接続すれば、ZigBee ネットワーク上に OSC メッセージを送ることで各種ディスプレイにメッセージを表示できる。この表示装置はバッテリー駆動可能なワイヤレス表示装置なので、使い勝手に優れている。

```
/e-ARK/hohno-01/Expr/Out/Display-NN
```

5.2 グループ S - 外部から ProtoZOA へ向けた情報の集約・収集

5.2.1 時刻同期

時刻情報が必要なアプリケーションのために、電源バックアップ機能を持たせた RTC (リアルタイムクロック) シールドを用意した。このシールドは、バッテリーではなく電気二重層コンデンサで RTC をバックアップしているため、長時間通電しないと電圧が低下しリアルタイムクロックが保時できなくなってしてしまう。このため、RTC シールドを使わない時には、専用のブレッドボードに挿して電源を供給するようにしている。現在、NTP サーバ、GPS、電波時計等からの正確な時刻取得は行っていないが、ProtoZOA 環境に正しい時刻を供給する体制をできるだけ早く整える方針である。最終的には以下のような名前で時刻を取得できるようにしたい。

```
/e-ARK/hohno-01/Expr/In/ToD/JST
```

```
/e-ARK/hohno-01/Expr/In/ToD/UTC
```

5.2.2 各種センサ

Arduino は、もともと各種センサや各種小型アクチュエータを接続、何かを計測したり何かを制御したりすることを得意としている。よって、ProtoZOA 環境でも Arduino にさまざまなセンサをとりつけ、取得した情報を ZigBee ネットワークに送出して共有したり、ZigBee ネットワークを介して受信した情報をもとにアクチュエータを動かす環境は早急に整備したいことのひとつである。前項で述べたように一部は OSC の名前空間における名前を設定しているが、各種センサ向けの名前の整備は今後の課題である。

```
/e-ARK/hohno-01/Expr/In/Sensor/...
```

ZigBee プロトコルによる通信を実現するために採用した XBee モジュールには、10bit A/D 変換器が 4 つ搭載されている。XBee モジュールの設定を変更することで、定期的に A/D 変換してその結果を指定した XBee モジュール (あるいはコーディネータ) に送信できる。そこで、Freescale 社のアナログ 3 軸加速度センサモジュールを取り付け、4 つの A/D コンバータのうちの 3 つを用いて 3 つの軸のそれぞれの方向の加速度を A/D 変換させることにした。XBee モジュール、加速度センサともに質量は 10g に満たないため、小型のり

チウムイオン電池やこれらを取り付けた基板を含めても 50g 以下で実装できた。この軽さゆえ、このセンサは簡単なクッションにくるんだ上で投げることができ、飛行中の加速度を把握できる。また手先に取り付けて腕を動かした場合にモーションを記録できる。これにより全身を使ったジェスチャーを ProtoZOA 環境に取り込み、なんらかの指示情報として利用できる。

ところで、加速度センサを取り付けた XBee モジュールが出力するのはただのデータであり OSC メッセージではない。OSC メッセージにするには、このデータを受け取った Arduino が OSC メッセージにすることになるが、たいていの場合はデータを受け取った Arduino でデータの受け取って処理するスケッチ (プログラム) が動いているはずなので、OSC メッセージ化にはあまり意味はない。なお、ZigBee ネットワークにおいて、メッシュネットワークが構成され、ある ZigBee モジュール A から別の ZigBee モジュール B にメッセージが転送される際、別の ZigBee モジュール C を経由する、モジュール C における転送は自動的に行われ、C を装着した Arduino 上のスケッチがメッセージの転送に気づくことはない。

5.2.3 入力装置

これら以外にも新たな入力装置が ProtoZOA 環境で利用可能である。現在、USB ホストシールドを導入し、キーボードやマウスといったパソコンではなじみ深い入力装置を ProtoZOA 環境で利用できるように調整を進めている。

5.3 グループ C - ProtoZOA 内部の情報の流通

メッセージが ProtoZOA 環境に到着するのでも、ProtoZOA 環境から発信されるのでもなく、短にメッセージが通過する状況も可能にしておく、異なる二つのネットワークを ProtoZOA 環境で橋渡しするといった使い方が可能になる。

ProtoZOA の "Z" を構成する ZigBee は、もともとメッシュネットワークを動的に構成できるので、既存のネットワークとの出入口を整備し、流入するデータが ZigBee ネットワークを溢れさせないようにすればよい。外部との境界点ではパソコン等と USB シリアル接続するか、有線イーサネット接続が主流である。後者については、IPv4 が主流である。これは、標準的なイーサネットシールドが、OSI 第 4 層以下のプロトコルスタックをファームウェアに焼き込んでいるからであり、IPv6 に対応させる余地がなかったからである。現在、OSI 第 2 層までしかファームウェアに焼いていない SPI 接続のイーサネットモジュールを入手し、これに contiki プロジェクトが開発した IPv6 プロトコルスタックを乗せるべく調整を進めている。

6. 非常時情報通信への応用

現在、すでに開発を終えた ProtoZOA に対応した上記の実装は、著者の研究室において日頃から動かし活用している。以下に挙げる機材もそうした実装の一部であるが、非常時情報通信への応用が期待できるので節を改めて言及する。

6.1 -ARK/Radio

-ARK/Radio は、Arduino シールドに以下の機能を盛り込んで実装した、多機能 FM ラジオである。1 枚の Arduino シールドには収まらないので、Arduino 本体を含めた 3 段構成にした。

- モノクロ LCD (英数字 16 文字 × 2 行、カラー LCD に変更する予定)
- プッシュスイッチ × 4
- 温度センサ (アナログ入力)
- 湿度センサ (アナログ入力)
- 気圧センサ (I2C で通信)
- ZigBee モジュール
- FM ラジオモジュール (I2C で通信)
- リアルタイムクロック (電源バックアップ付、I2C で通信)

この ProtoZOA 実装の要は、FM ラジオモジュールである。このモジュールは、AIROHA 社の AR1010 (Single Chip FM Stereo Receiver) を用いており、FM 局の選択や音量の調整などを I2C を介して制御できる。日頃は、時刻、温度、湿度、気圧などを表示する時計として、あるいは普通の FM ラジオとして利用できる。普通の FM ラジオや時計と異なるのは、ProtoZOA なので、ZigBee モジュールを搭載し OSC メッセージを受信できることである。すなわち、-ARK/Radio は外部から遠隔操作可能なラジオである。これにより、緊急地震速報や緊急警報放送が発せられ FM ラジオ局で音声による説明が始まった場合、それを伝える OSC メッセージを ZigBee ネットワーク上でアナウンスすれば、これを受信した -ARK/Radio は、ただちに FM ラジオを音量最大で起動できる。インターネットが普及した現在であっても、非常時にラジオ放送が果たす役割は未だに大きい。

現時点では未実装だが、プッシュスイッチを操作したり、加速度センサを追加して -ARK/Radio を一定のパターンで叩くなどした場合に、特定のメッセージを時動的にツイートしたり、あらかじめ決めた相手にメッセージをメールしたりする機能も付加できる。プライバシーとの兼ね合いもあるが、同意の下で利用すれば非常通報装置としても利用可能であ

る。なお、上記を実現するためには、当然ながら何らかのインターネット接続と、インターネット接続と ZigBee ネットワークを結ぶゲートウェイが必要になる。

6.2 -ARK/ALERT

-ARK/Radio は、非常時には緊急地震速報や緊急警報放送の開始を知らせる OSC メッセージを受信して起動する。この OSC メッセージをどのようにして発するかにはいくつかの方法が考えられる。インターネット上で情報を検索したり、インターネット上を流れるメッセージを受信して緊急地震速報や緊急警報放送の開始を知る方法もあるが、非常時の種類によっては、これらの放送が始まることをインターネット上で察知するより前にインターネット接続が途絶する場合も考えられる。

このような場合に対応する方法として、-ARK/Radio 自身が緊急地震速報や緊急警報放送の開始を常時監視する方法もあるが、-ARK/Radio が 1 台しかない環境でこれを行うと、-ARK/Radio の利用者が FM ラジオとして自由に利用することができなくなる。緊急地震速報や緊急警報放送は、全ての FM 局で実施しているわけではないからである。

対処方法としては、-ARK/Radio を複数台用意し、そのうちの 1 台が緊急地震速報や緊急警報放送を常時監視し、放送開始を検知したら他の -ARK/Radio に OSC メッセージで伝達するという方法がある。あるいは、市販の「緊急地震速報・緊急警報放送受信機」を導入し、この受信機が放送開始を検知して動作を開始したら、これをトリガにして OSC メッセージを発する ProtoZOA 実装を作ることである。二つの方法には一長一短がある。前者はハードウェアの実装は可能であるが、緊急地震速報や緊急警報放送を正しく受信できるかを確認することが現状では難しい(正しく受信できるコードが得られれば最適な解である)。後者は、緊急地震速報・緊急警報放送受信機を導入した上で、緊急地震速報や緊急警報放送の開始を検知するためのアダプタが必要になる。現在国内で入手できる緊急地震速報と緊急警報放送の両方に対応した製品は、ユニデン製の EWR200 しかなく EWR200 には緊急地震速報や緊急警報放送開始を出力する端子等はない。そこで、緊急地震速報や緊急警報放送開始の際に点滅する受信機本体表面の LED に光センサを張り付けて LED の点滅を検出することにし、すでに実装した。

6.3 -ARK/L7GW

非常時においてある地域が激しく被災した結果、一時的にインターネット接続性を失ったとする。この場合、誰かがスマートホンを使って 3G 回線等を経由してインターネットにアクセスして対外接続性を確保した上で、即席の無線 LAN アクセスポイントを構成すれば、多くの WiFi 対応端末利用者は無線 LAN アクセスポイント経由で対外接続ができるので、

情報が全くやりとりできない事態は回避できる。しかし、最近ではテザリングと呼ばれることもあるこの方法で対外接続を確保できたとしても、多くの人間に限られた回線を共有するとひとりあたりの通信速度の低下は明白で、何らかのフィルタリングを施す必要がある。ところで、これまでは 3G 回線等を経由した対外アクセスができなければ、対外アクセスはあきらめざるを得なかったが、3G 回線等で対外アクセスを行う通信装置と無線 LAN アクセスポイントの間を ZigBee ネットワークで橋渡しできれば、3G 回線等へのアクセスは避難場所から離れた電波状態のよい場所を実施すれば問題は解決できるかもしれない。

ZigBee ネットワークでは、コーディネータとエンドデバイス間の距離が長く直接通信できなくても、両者の間に 1 台あるいは複数台のルータデバイスを配置できれば通信は自動的に中継される。上述の被災者が集まっている場所では電波状況が悪く 3G 回線等にアクセスできないという仮定において、3G アクセスが可能な場所と被災者が集まっている場所を ZigBee ネットワークで橋渡しするのはアイデアとしては面白いがいくつかの問題を解決する必要がある。

データ通信向けにしてはとても遅い ZigBee の通信速度に配慮し、意図せず大きなデータが流れないように、ZigBee による橋渡しの両端でコンテンツをフィルタリングする必要がある。すなわち第 7 層でのゲートウェイが必要となる。メールのやりとりやツイートのようリアルタイム性が不要のないのなら、ここにかつての UUCP のようなストア&フォワード型の通信プロトコルを用いるのも解決策のひとつとなる。

ZigBee ネットワークの実質的な通信速度は 25kbps ~ 50kbps 程度であるが、20 年以上前は、大学全体の対外接続速度が 19.8kbps 程度だったこともあるし、10 年程度前は個人のモバイル通信環境で 32kbps という速度は珍しくなかった。現在のマルチメディアデータや大規模なソフトウェアアップデートを一次的に中止し、真に不要不急なトラフィックだけを通せば ZigBee ネットワークによるネットワーク中継の可能性は十分にある。

ZigBee ネットワークによるネットワーク中継の別の可能性は、その軽さに見出せる。ZigBee モジュールと電池とアンテナがあれば中継ができる。電池の大きさにもよるが、1 時間程度の中継であれば、総重量は 100g 以下にできる。電池を少し小型にすれば、80g 程度で ZigBee 無線中継ノードが完成する。実験はこれからであるが、ヘリウム入りの小型の風船が 4 つ程度あれば、中継ノードを空中に浮かべることが可能である。風船を浮かべると高さによっては、風船と地上の構造物と結びつけるための糸の重さも無視できないし、安定した浮力を得るためには 4 つの風船で十分かは未確認だが、ZigBee 中継ノードを空中に浮かべる実験も近日中に実施する予定である。

7. 評価と考察

ProtoZOA アプローチでは、ハードウェアを Arduino と Arduino シールドで実現し、通信の下位層を ZigBee に委ね、上位層は OSC を使うことで短時間で環境構築が実現できる。

これらは、Arduino のように GPL やクリエイティブコモンズライセンスで公開されていたり、ZigBee のように標準化されていたり、OSC のようにそれに準じるデフォルト状態になっていたりとするので、開発に際して必要になる多くの情報が容易に取得できるという特徴がある。これは著者らだけでなく、第三者が同じ設計方針で ProtoZOA と同じか類似した環境作成する際にも有益な点である。

ProtoZOA に至る前の段階では、“O”の部分に独自プロトコルを用いていた。独自プロトコルには実装者の好きのようにできるというメリットがあるが、多数のプラットフォームの多数の言語に実装することは難しい。一方、OSC は、単純な名前空間にデータ型と値を割り当てるといったシンプルな方式で汎用性が高く、すでに多数のプラットフォームの多数の言語に OSC 対応ライブラリが提供されている。OSC 採用にはこの点が決め手となった。広義の“A”の対象である Apple iPhone/iPod Touch/iPad (以下、iPhone 等) や、Google Android デバイスにも OSC メッセージを送受するアプリケーションはいくつも登録されている。狭義の“A”である Arduino と広義の“A”である iPhone 等や Google Android デバイスが OSC メッセージを交換できることを確認する際には、WiFi 環境上の広義の“A”と ZigBee ネットワーク上の Arduino が OSC メッセージを交換するためのゲートウェイが必要になるが、それだけでメッセージ交換が可能になった。もし独自のプロトコルなら、広義の“A”デバイス全てに独自プロトコルを実装しなければならない。

OSC は、MIDI の後継として開発されたため、MIDI との親和性が良好である。この特徴は、ProtoZOA 開発当初は「ProtoZOA 環境から MIDI デバイスも操作可能である」という程度のものであったが、iPhone 等の OS が iOS 4.2 になって、当該 OS が MIDI を正式にサポートするに至って状況が変わった。iPhone 等では、外部との情報交換は WiFi にほぼ限定されている。Bluetooth は、SPP (Serial Port Profile) のサポートがないため、ユーザレベルでの自由な情報交換には適さない。拡張ポートには、シリアルポートが存在するが Jail Break という特殊な方法を用いない限り利用できない。Jail Break は個人的なチャレンジの範囲内であれば興味深いが Jail Break を前提とした研究開発は受け入れられない場合がある。オーディオ入出力を介して、低速のアナログモデムをソフトウェア的に実現した例がありこれはこれで有力ではあるが、必ずしも安定動作するわけではない状況にあ

る。これに対して、iOS が MIDI をサポートした結果、iOS 上で MIDI プロトコルの送受が可能になったので、MIDI を介して OSC メッセージをやりとりする道が開けた。MIDI を用いて iPhone 等を ProtoZOA 環境に取り込む作業は今後の大きな課題である。

現時点では、ProtoZOA の性能評価や安全性評価は十分ではない。

ProtoZOA アプローチは、採用している技術に起因する問題点がいくつかある。たとえば、ZigBee を用いているため、PAN ID が偶然衝突した場合や、悪意の第三者が PAN ID をぶつけてきた場合にはセキュリティ上の対処が難しい。PAN ID が判れば、当該ネットワークのコーディネータとはいつでも通信できるので、DDoS 攻撃が可能である。PAN ID がわかってしまい、OSC メッセージの名前構成が判れば、特定のデバイス进行操作できてしまう。また、混雑している 2.4GHz 帯で ZigBee が数多く参入した場合、安定して利用できるかも調査しなければならない。このあたりは、ProtoZOA アプローチの根幹に係わることなので、現在確認作業を進めている。

8. おわりに

本報告では、ProtoZOA アプローチを提起し、当該アプローチに基づいて設計実装された機材について報告した。また、すべての機材は、著者らの研究環境の改善に寄与し、いくつかの機材は、著者の研究テーマである非常時における情報通信環境の改善にも寄与することが確認できた。本報告は、ProtoZOA アプローチの提起を中心に記述した。詳細な定量評価や考察は次報以降で行う。

謝辞 本研究のうち、-ARK デバイスに関わる部分は、総務省戦略的情報通信研究開発推進制度 (SCOPE) 地域 ICT 振興型研究開発案件として平成 21 年度に新規採択されたプログラムに基づいて実施したものである。総務省および同省北陸総合通信局の関係各位に深謝する。

参考文献

- 1) : ZigBee Alliance. <http://www.zigbee.org/>.
- 2) : opensoundcontrol.org. <http://opensoundcontrol.org/>.
- 3) Team, A.: Arduino Home Page. <http://www.arduino.cc/>.
- 4) : The Operating System for Connecting the Next Billion Devices - the Internet of Things. <http://www.sics.se/contiki/>.
- 5) 猪俣敦夫, 大野浩之: 乾電池でも運用可能な「非常時対応電子アーミーナイフ」(-ARK) を用いた非常時情報通信システムの実装, *Internet Conference 2008*, pp.15-24 (2008).
- 6) 株式会社アクセンス・テクノロジー: StewGate 1 を使う. <http://stewgate.appspot.com/>.