

格子状ネットワークにおけるグローバル通信 とローカル通信を組み合わせた ユビキタスコンピューティング

塚本昌彦[†] 藤田直生[†]

ユビキタスコンピューティング環境では、環境に埋め込まれた多数の小型のコンピュータをトポロジに基づいて全体を高度に制御することが求められる。本研究では格子状のネットワークを構成するためのデバイスを設計・実装した。格子状のネットワークはローカルなトポロジとグローバルなトポロジの対応が自明であり、ローカルプログラミングの組み合わせによりマクロなプログラミングを実現できるため、有用である。本稿ではこのデバイスを用いて、グローバル通信とローカル通信を組み合わせることで高度なコンピューティングを可能にするいくつかの手法について述べる。

Ubiquitous Computing Combining Global and Local Communication On Grid Topology Network

Masahiko Tsukamoto[†] and Naotaka Fujita[†]

In ubiquitous computing environments, there is a strong demand for advanced methods for controlling a lot of small computers embedded in the environments as a whole based on their topology. In this research, we designed and implemented a device for constructing a grid topology network, which is useful for our purpose because local topology is directly mapped into the global topology and macro-programming is easily achieved by combining local programming. In this paper, we show several methods for achieving macro-programming by combining global and local communication on this device.

1. はじめに

近年、IC タグやセンサなどを用いたユビキタスコンピューティングが生活に浸透してきており、近い将来、人々の暮らしの中で多数の超小型コンピュータが生活環境内に埋め込まれて広く使われることが見込まれるようになってきている。しかしこれらをうまく分散協調動作させるための技術は現在のところ不十分である。広く使われている MICA/Mote, Arduino, Gainer, Sun Spot などは一般に個々のデバイスに対するプログラミングが必要であり、多数のデバイスを連携させるためには複雑かつ高度なプログラミングが求められる。

本来望ましいとされるユビキタスコンピューティング環境とは、多数のユビキタスコンピュータがどのように動作しているかはユーザ側からは見えず、その存在を意識することなく利用できるというものである。そのため、コンピュータ単体で行う処理よりもコンピュータ全体で何ができるかということが重要であり、効率よく多くのコンピュータを全体として制御するメカニズムが必要となる。また、多くのコンピュータを扱うために、ユーザによる入力がなくとも、コンピュータ同士が自律的に連携を取り合い、便利・快適、安全・安心、豊か・楽しさ、エコ、健康などといった全体的な目標に向かって個々のデバイスが協調して動作を行うことが望ましい¹⁾。

これに対し筆者らの研究グループでは、これまでに多数のユビキタスデバイス全体としてプログラムする『全体プログラミング』の研究を行っており、さまざまな手法やシステムを開発してきた^{2)~7)}。具体的な研究目標は以下のようなものであった。

- ・デバイス全体を一斉に動作させ、同期を取りながらセンシング、I/O 制御を行う。
- ・地面にばら撒かれたセンサをうまく交代で動作させて動作持続時間を長くする。
- ・空間にばら撒かれた発光デバイスを使って大きなスクリーンとして利用する。
- ・自律移動可能なユビキタスデバイスを用いて、全体を整列させ、一列になって移動させる。あるいはアリの採餌行動などを模して、特定のものを探したり、運搬したりさせる。

類似したアプローチとしてユビキタスデバイスのマクロプログラミングに関する研究がある^{8)~12)}。しかしこれらは個々のデバイスのプログラミングの延長であり、個別の ID を指定する必要があるなどデバイスの数に対するスケーラビリティがないのが現状である。上記のような目標において、多数のデバイスを連携動作させる『群コンピューティング』を実現するには、個々のデバイスがどのようにつながっており全体の中でどのような位置にあるかを知り、それをうまく利用してそれぞれの位置に応じた役割を果たしていくこと、すなわちトポロジをうまく利用したプログラミングが重要である。

[†] 神戸大学
Kobe University

本研究ではユビキタス環境において多数のユビキタスデバイスによる群コンピューティングを実現することを目標とする。すなわち、環境内に数百から数千個規模のユビキタスデバイスが存在することを想定して、環境内のデバイス全体を対象としたプログラミングを行えるような方式を確立する。今回特に、ネットワークのトポロジを用いてうまくコーディングしていくこと（トポロジコーディング）により、プログラミングのなかで全体を描くことができるような基盤を構築することを考える。

そこで本稿では、ローカルとグローバルのトポロジの対応が比較的自明な格子状有線ユビキタスネットワークを対象としてこれを実現する。格子状ネットワークではホップをカウントすれば座標値が得られる。グローバルとローカルな通信を併せ持つシステム（グローバルグリッドと呼ぶ）を構成し、グローバルグリッドを管理するためのシステムアーキテクチャを考える。すなわち、接続性のチェックやコンフィグレーションができるようにする。

2. 格子状有線ユビキタスネットワーク（グローバルグリッド）

2.1 グローバル通信とローカル通信

多数のコンピュータ間で通信を行う際、グローバルに通信を行うかローカルに通信を行うかは重要な選択となる。本稿では有線の通信を考えるが、無線通信の場合には通信範囲が広い場合がグローバル通信、狭い場合がローカル通信に対応するものと考えられる。

一般に、グローバル通信とは全てのノードに対して行う通信のことをいい、ノード数が増えた時のスケーラビリティがある。全てのノードに一斉に同じ情報を伝えるのに効率のよい方法である。また全てのノードの時刻的な同期をとるのにも向いている。一方、多数のノードが送信を行う場合には、送信タイミングの衝突回避が困難となるため、通信効率が悪くなるという問題がある。基本的に一つの回線を多数のノードが共有することになるため、多数のノードができるだけ同時に送信を行わないように、通信手順の設計上十分な配慮が必要となる。

ローカル通信とは近隣のノードのみと通信を行う手法である。多数のノード間で情報を共有したい場合や近接にないノードに情報を送信したい場合には中継を行う必要があり、通信に時間とリソースのコストがかかることになるが、近隣だけの情報交換を目的とするなら効率のよい通信手段である。通信が干渉しない範囲で多数のノードが同時に送受信できるため、個別の通信を各所で行ううえではスケーラビリティがあり、並列性ももつ。しかし、離れた場所にあるノード間や多数のノード間でタイミングの同期をとるのは難しい。互いに通信ができるかどうかは物理的な距離を表すものと考えられるため、通信可能関係をグローバルに解析すれば、ネットワーク全体の地理的關係を抽出できるという特殊な機能も有する。

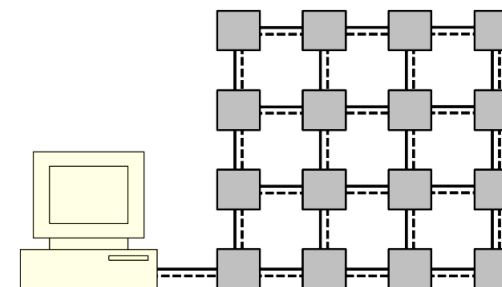


図 1 グローバルグリッドのトポロジ例

両者は通信の特性として相補的な関係にあり、ユビキタス環境においては両者を必要とするケースが数多く存在する。本稿では両者を物理的に兼ね備えるネットワークを考える。

2.2 グローバルグリッドの概要

本稿では、グローバル通信とローカル通信の両者を備えるネットワークを考えるうえで、ローカルなトポロジとグローバルなトポロジの関連性が自明な、格子状のトポロジに限定することとした。農場などでセンサや LED を用いたセンサネットワークを利用する際、継続的な電源供給を行わなければならないことから、格子状の有線ネットワークを考えることは奇異なことではない。このようなネットワークをグローバルグリッド（Glocal Grid）と呼ぶ。グローバルとはグローバルとローカルという言葉を組み合わせた言葉で、通信の文脈ではよく使われる言葉である。以下ではグローバルグリッドの様々な側面について考察していく。

本稿で考えるグローバルグリッドの典型的なトポロジ例を図 1 に示す。この図では 16 個の四角形がノードを表し、マイコンやセンサ、LEDなどを備えたユビキタスコンピュータを意味する。各ノードは縦横方向に規則正しく並んでおり、図では実線と破線で表される二種類の回線につながれている。実線はグローバル通信を表しネットワーク全体でつながっている。破線はローカル通信を表し隣接間のみでつながっている。二種類の回線に対して必要に応じてホストコンピュータからコマンドを送信したり、情報を収集したりするため、ノードの一つがこの二種類の回線を介してホストコンピュータに接続されている。図では左下のノードがパソコンにつながれている。一般にホストコンピュータやネットワークにつながるノードはシンクノードと呼ばれる。通信の方式としては、グローバル、ローカルともに、一般的なシリアル通信を想定する。

各ノードには N ビットの識別子（ID）が付与されているものとする。特に断らない限りそれぞれのノードには一意の ID が付与されていることとする。

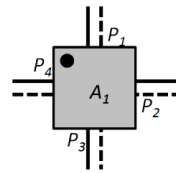


図 2 グローカルグリッドノードの構成

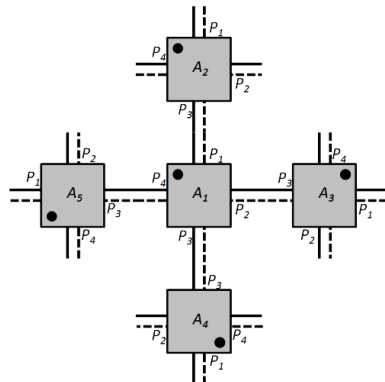


図 3 隣接関係の抽出

2.3 ノードの構成

図 2 に示すように、各ノードには上下左右の 4 方向に対してケーブルをつなげるコネクタを有する。コネクタにはローカル通信およびグローバル通信の接続部がある。便宜上コネクタに対応して、内部のローカルポートをある一つのものから時計回りに P_1, P_2, P_3, P_4 と名付けておく。図中のノードには、必要に応じて、 P_4 と P_1 の間のコーナーに黒丸の印を記す。ノード A からグローバルおよびローカルにメッセージ M を送出することを、それぞれ、

A,G: M

A,L: M

と記す。ローカル通信は上下左右の 4 方向に対して同時に送出されるものとする。すなわち、A, L:M は A に隣接するすべてのノードが M を受信する。

各ノードは受信したメッセージがグローバルメッセージかローカルメッセージかを受信時に区別できるものとし、さらにローカルメッセージの場合、どのポート (P_1, P_2, P_3, P_4) から受信したのかを区別できるものとする。ホストコンピュータからメ

ッセージ M を送出する場合は、

G: M

L: M

と記す。接続されたノードはメッセージ M を受信する。ホストコンピュータは ID などによって区別できるようになっているものとする。

3. グローカルグリッドに対するオペレーション

3.1 基本設定

2 章で述べたグローバルグリッドに対するさまざまなオペレーションコマンドを用意する。順に説明していく。

(1) ID 設定

ノード ID をランダムに設定させるコマンドを考える。ホストコンピュータからグローバルにコマンド `random_id` が送出されて使われるものとする。具体的には

G: `random_id`

G: `random_id(id)`

G: `random_id(x,y)`

のいずれかを受信するとノードはランダムに自局 ID を割り当て直す。最初のコマンドは全てのノードに対するもの、中央のコマンドは特定の ID を持つノードに対するもの、最後のコマンドは後で述べる特定の座標のものに対するものである。高性能な乱数機能を持たないマイコンなどでは、立ち上がりからの時間やアナログピンの入力値などを用いる手法が知られているが、乱数の種はそれぞれのノードにあらかじめ設定しておくことが望ましい。

(2) 隣接関係抽出

ホストから隣接関係抽出のコマンドが出されると、各ノードがローカルメッセージとして自分の ID を隣接ノードに対して送信することで互いにどのポートにどのような ID のノードが接続しているかを知る。ローカルメッセージは隣接ノードのさらに隣接する自局以外の 3 ノードからのメッセージと衝突する可能性があるため、ランダムジッタを入れて送信する必要がある。4 ノードの衝突を考慮するためには 20 スロット程度を用意してランダムにスロットを選んで送信するのがよい。初期状態ではどのようなノードがどのようにつながっているかわからないので、衝突を避けるためにランダムジッタを使用することは必須である。後で述べる座標値を先に決めることができるなら、衝突を避けるようにグローバルにメッセージを発信するノードを指定することで効率よく情報交換が行えるが、今回の前提条件では隣接関係を抽出してからでない座標値を割り当てるのが難しいため、このような手順としている。

図 3 の状況下では、ホストからコマンド

G: `neighbor_config`

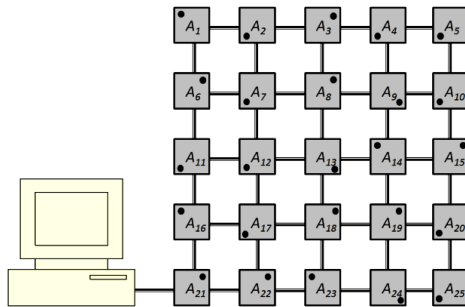


図 4 方角と座標の設定

を受けると、図中の5つのノードはランダムジッタ後に

$A_2, L: 2$

$A_3, L: 3$

$A_1, L: 1$

$A_5, L: 5$

$A_4, L: 4$

などのようにメッセージを送出する。 A_1 はポートごとに受信したIDを、例えば、

$P1: 2, P2: 3, P3: 4, P4: 5$

のように記録する。以降ローカル通信においては、メッセージの先頭に宛先IDを付与することで、4方向へのブロードキャストではなく特定ノードへのメッセージとすることとする。例えば、

$A_1, L: 2 M$

は A_2, A_3, A_4, A_5 が受信するが、 A_3, A_4, A_5 は破棄し、メッセージが自局IDで始まる A_2 のみがメッセージMとして受信する。これを、

$A_1 \rightarrow A_2: M$

と略記することとし、「 A_1 から A_2 にMを送信する」という言い方をする。隣接関係抽出が完了しており、しかもトポロジが変化していないという状況において使用する。

先の例と同様、

$G: neighbor_config(id)$

$G: neighbor_config(x,y)$

というようなコマンドも有効である。特定ノードにだけローカルなメッセージを送出させることになる。グローバルな指示を受けなくても、トポロジの変化やスリーブからの立ち上がり時などには隣接局に対してローカルで

$A_1, L: neighbor_config$

を送信することもありうる。この場合隣接4ノードからIDが送信される。

(3) 方角の設定

ノードを接続した場合、一般に図4に示すように各ノードはさまざまな方向に接続されることになる。トポロジを考慮したユビキタスコンピューティングを可能にするためには、方角と座標の設定が重要になる。方角は、便宜上 North, South, East, West に対応して N, S, E, W の記号を用いるが、地理上の方角に一致する必要はないものとする。ここでは(1)の隣接情報が設定されているものとし、ローカルに順次メッセージを伝搬することで実現できる。まず、パソコンから

$L: compass_config(E)$

が送信される (Eとしたがこれは何でもよい)。受信したノード A_{21} は A_{22} に E を、 A_{16} に N を送信する。つまり、

$A_{25} \rightarrow A_{22}: compass_config(E)$

$A_{25} \rightarrow A_{16}: compass_config(N)$

が送信される。これを順次繰り返す。N,E,S,W のメッセージを受信すると、受信ポートの反対側に同じメッセージを転送し、時計回りに方角を一つずつ増やして転送することになる。一般にポート i から $compass_config(d)$ (d は N,S,E,W のいずれか)を受信すると、ポート $i+2+j \bmod 4$ から $compass_config(d+j \bmod 4)$ ($j=0,1,2,3$)を送信する (N,S,E,W を $0,1,2,3$ と同一視している)。ノードは N の方角をポートの番号で記録しておく。メッセージのループを防ぐために、受信した方向には返さないこと、既に (一定時間内に) 方角が設定されている場合には転送しないこととする必要がある。以降、ノード A から N(S,E,W)の方角に (ローカルに) メッセージMを送出することを

$A N: M$

と略記する。前述と同様、設定の完了とトポロジ不変化を想定して使うものとする。

(4) 座標の設定

座標(x,y)の設定は方角の設定とほぼ同様の手順で行うことができる。

$L: coord(0,0,1,0)$

これは受信したノードの座標が(0,0)であり、進行方向が(1,0)であることを示す。この場合、Eの方向がX軸、Nの方向がY軸となって座標系が形成されることになる。転送の手法やループ防止のメカニズムは方角設定と同様である。最初の値は(0,0)以外にすることも可能である。座標値は負の値になりうるが、初期値の設定により調整できる。座標値はノード内に記録しておく。座標値が決まった後、

$G: assign_id$

を実行すると、座標値からあらかじめ決められた方法でIDを生成する。IDから座標値の計算方法もあらかじめ分かっているものとする。ローカルにアドレス解決ができるというメリットがある。

3.2 データ通信

(1) 個別通信

ノード間のデータ通信はグローバル通信で行うか、ローカル通信のマルチホップで行うことができる。マルチホップのルーティングは、座標を用いて自明な方法で行うことができる。図4でA₁₀（座標は(4,3)）が座標(1,1)（A₁₇）にメッセージMを送信したい場合を考える。グローバル通信を用いる場合は以下のように送信する。

A₁₀, G: *message(src:id(10),coord(4,3), dst:coord(1,1), data:M)*

A₁₀, G: *message(src:id(10),coord(4,3), dst:id(17), data:M)*

ローカル通信を用いる場合は座標がより近いA₁₅に転送する（A₉を選んでよい）。

A₁₀→A₁₅: *message(src:id(10),coord(4,3), dst:coord(1,1), data:M)*

A₁₅は宛先（dst）を見てより近い方へと転送してゆく。

A₁₅→A₁₄: *message(src:id(10),coord(4,3), dst:coord(1,1), data:M)*

これが繰り返される。宛先ノードのIDがわかっていて座標がわからない場合はグローバルに問い合わせを発する。

A₁₀, G: *query(id(17), coord?)*

これに対してA₁₇が答える。

A₁₇, G: *answer(coord(1,1))*

これに続けて先ほどのメッセージを送ることができる。

(2) データ集約

多数のノードからデータを集めるのは一般に難しい問題である。個々のノードが持つセンサデータなどをパソコンで全部収集するような例を考える。単純な方法としては、座標値を用いてノードを一つずつ指定し、グローバル通信で答えさせる方法が考えられる。ノード数の数に比例した時間がかかる点が問題である。ローカル通信を用いて一番通り所から1ホップずつシンクノードに向けてデータを送っていくという方法も考えられる。衝突を回避するためにグローバルに制御信号を送りながら実行するのが有効かもしれない。複合的な手法として、いくつかのノードにデータを集めてから答えるノード数が定数オーダーにしてからブロードキャストで答えさせる方法が考えられる。データの処理を伴う場合には有効である。データ集約については通の関連研究があるので今後より詳細な検討が必要である。

(3) 同時個別通信

同時個別通信はグローバルグリッドに向けた通信形態である。例えば全てのノードがN側のノードに自局のセンサ値を送るというような通信である。ホストからそれを指示することも考えられる。

G: *send_req(all, N, X)*

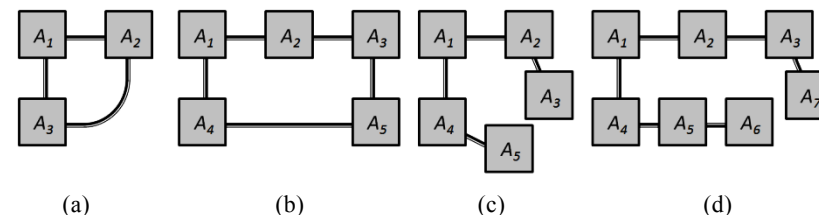


図5 形成不良の例

allは全てのノードに対する指示であることを意味するものとする。これに対し、

A_i N: *message(src:id(i), X)*

のような送信が行われる。ただし、今回の前提条件ではローカル通信の競合が起こり、いくつかのノードは上下方向からのデータ送信を同時に受信することになる。そのためホストは、

G: *send_req(6..10, N, X)*

G: *send_req(11..15, N, X)*

G: *send_req(16..20, N, X)*

などのように複数回にわたって指示しなければならないかもしれない。ここで1..5はID1からID5までの5つのノードに対する指示を意味する。

結果はLEDを用いて出力し、カメラなどでグローバルに同時に取得することが考えられる。アドレスの重複検出やデータ集約などもこの方法を用いれば簡単に行うことができる。これらの点についても今後のより詳細な検討を要する。

3.3 さまざまなアルゴリズム

(1) 形成不良検出

一般にネットワークは正しく形成されているとは限らないため、形成不良（ill-formed）を検出できるようにすることは有用である。図5に形成不良の例を示す。

(a)は方角設定、座標設定で、(b)は座標設定で検出できる。矛盾するメッセージを受け取ったノードがグローバルにメッセージを送出するのがよい。

(c)(d)は、座標割り当てをした倍と同じ座標を持つことになるノードが二つ以上ある例である。一般に方角設定、座標設定では検出はできないが、隣接情報を隣接ノード間で交換することで、(c)の場合はノードA₁がA₂とA₄の隣接情報を比較することで検出できる。(d)の場合はさらに広い範囲で隣接情報を交換しなければならない。次の欠損リンクの場合と同様であるのでそちらで可能な手法について議論する。

(2) 欠損リンク検出

欠損リンク（missing link）の例を図6に示す。左ではA₃とA₄の間のリンク、右ではA₂とA₅の間のリンクが欠損している。ユーザの意図によるものである場合も考え

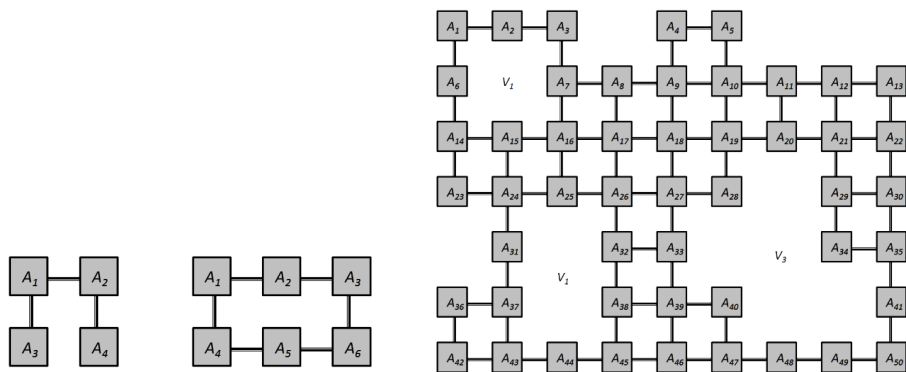


図 6 欠損リンクの例

られるが、ケーブルの接続不良による場合もあり、検出できれば有用である。(e)においては v_1, v_2, v_3 のようなリンク欠損部がある。特に v_2, v_3 は大規模な欠損部であり、ルーティングに支障をきたす恐れがある点に注意が必要である。例えば A_{46} から A_{34} にメッセージを送信する場合に A_{40} に到達してしまふと行き止まってしまう。このようなルーティングの行き止まりのことは一般に「ボイド」と呼ばれ、一般にルーティングにおけるボイド回避の方法はよく知られている。メッセージに「ボイド回避」のフラグを立てて、右回りで欠損部を縁に沿って巡回していく(左回りでもよい)。最悪外周をたどってしまうことになる(例えば A_{49} が存在しないのに右回りの回路を選んだ場合など)が到達性は保証されるという意味で有効である。

ルーティングのボイド検出はルーティング時に動的に行うことができるが、事前にこのような部分を見つけておくことも有効である。内部のノードの電源が落ちている場合や物理的な損傷などによるものである場合が考えられる。基本的に4隣接の一つでも欠損しているノードがからスタートし、欠損方向を意識しながら右回りか左回りでメッセージを転送していくことにより検出する。自ノードの ID を順次メッセージに付け加えていく必要があるが、パスが長くなった場合にはルートの途中のパスを省略してホップ数だけを記録することも可能である。

4. 実装

4.1 ハードウェア

グローバルグリッドシステムにおけるノードの構造を図7に、仕様を表1に示す。実装したデバイスの外観は図8に示す。計算機には、8ビットマイコンである Atmel

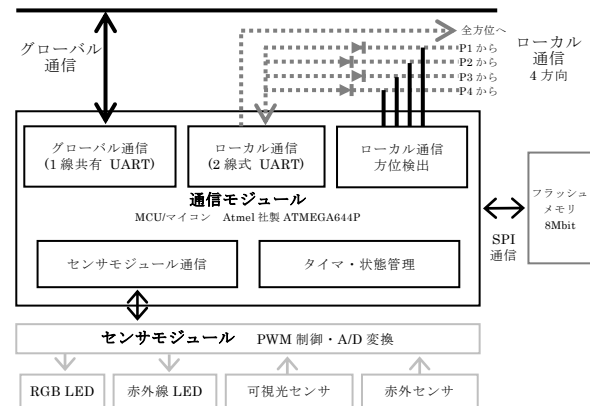


図 7 グローバルグリッドデバイスの構造

表 1 グローバルグリッドデバイスの仕様

MCU/マイコン	Atmel 社製 ATMEGA644P Flash プログラムメモリ 64kB EEPROM 2kB SRAM 4kB
フラッシュメモリ	8 Mbit (1 MByte)
グローバル通信	TTL シリアル通信 (UART) 115.2kbaud 1線共有方式 半二重通信
ローカル通信	TTL シリアル通信 (UART) 115.2kbaud 全二重通信
光センサ	可視光センサ A/D変換 10bit 赤外線センサ A/D変換及びデジタル入力
LED	RGB フルカラーLED 各色 8bit PWM 制御 赤外線 LED
電源	3V 動作 (1.8V~5.0V)

社製 ATMEGA644P を採用した。データ保存用として、外部にフラッシュメモリを 8Mビット有しており、センサ情報などを保持しておく。メインとなる計算及び通信システム用とセンサ用は別モジュールとして設計しており、計算及び通信システム用の通信モジュールはすべてのノードで同じ仕様とし、センサ用モジュールは環境に合わせて変更できるように実装した。センサは LED や光センサを搭載する以外に、加速度センサや温度センサなどを搭載することが想定されることから、通信モジュールとは分離して取り付けが可能である。標準的なセンサ用モジュールは、RGB フルカラーLED

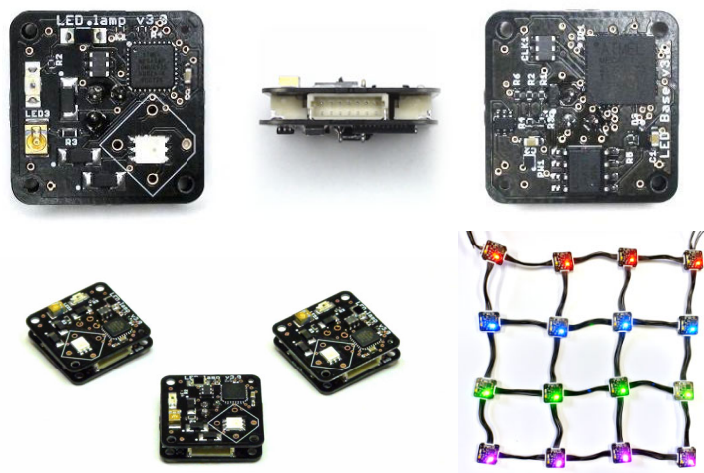


図 8 実装したデバイスの外観

と赤外線 LED, 可視光センサと赤外線センサを搭載しており, 光に特化した環境センサとした。これは人との連携には光情報がわかりやすいためである。また, 可視光及び赤外線の発光と受光ができることから, 光による制御及び連携が可能である。図 8 に示す外観では, コネクタを中心に挟む 2 つのモジュールで構成されている。

グローバルグリッドのノード間は有線で接続しており, 各デバイス間は電源線及び通信線をあわせた 7 線をコネクタにより接続している。グローバル通信は 1 線による半二重通信とし, ローカル通信は 2 線による全二重通信で接続される。

4.2 通信仕様

(1) グローバル通信

グローバル通信は, 1 線共有の半二重通信でのシリアル通信である。各ノードは受信状態で常時待機しており, 必要に応じて送信を行う方式で, 各ノードが共有通信路を他のデバイスが使用しているか判断してから通信を行う。通信速度は 115200baud で通信しており, 設定に合わせて変更可能である。さらにノードからではなく, パソコンからの通信は USB を介して可能である。

全体で情報を共有できることから, ノードのトポロジに関わらず通信経路が確保されるという特徴がある。また, 特定のノードに対しての通信は, 各ノードに設定された ID を指定することで行う。

通信メッセージはテキストベースとし, Newline までの一行を 1 メッセージとして認識する。ID 指定は, 16bit の任意の番号を使用し, 「111:」形式で行頭に付けること

で認識される。また, 65535 番は未設定 ID として扱われ, ID 指定が無い場合はすべてのノードに対して有効なメッセージとして認識される。

(2) ローカル通信

ローカル通信は, 2 線式の全二重通信のシリアル通信である。グローバル通信とは違い各ノードから送受信が同時に可能である。通信速度は 115200baud で通信しており, 設定に合わせて通信速度を変更可能である。近接 4 方向のノードとのみ通信でき送信情報は 4 つのノードで共有される。また, どのノードから通信されているかを認識するため, 図 7 に示すように通信とは別に各通信線を監視する方式をとっている。これにより接続された 4 方向からの通信を認識すると同時に同時通信による衝突検出を行える。

通信が近接のみで共有されることから, ID を指定しなくとも通信可能であるが, 送信情報が 4 つのノードで共有されることから ID を指定することが望ましい。また, 通信方向を特定できることから, トポロジに依存した通信経路の構築が可能であり, 通信のルーティングを実験することができる。

通信メッセージはグローバル通信と同じく, テキストベースとし, Newline までの一行を 1 メッセージとして認識する。ローカル通信では 4 方向に接続されているノードを認識できるため, 通信初期化後は各ポートに接続された ID 情報を保持する。初期化時に接続情報が確認できなかった場合は 65535 と認識し, 未接続として扱う。この情報を元に, 通信方向の指定やデータ転送方向を指定することが出来る。通信コマンドは, グローバル通信と同じく 16bit の ID を先頭に付けて通信を行うか, ID 無しによるノードを特定せずメッセージを送信する方式がある。

4.3 センサモジュール

センサモジュールは通信モジュールから独立しており, 光センサと LED を備えている。通信モジュールとは 1 線共有式の UART により通信しており, 通信により LED 及びセンサの情報を共有し, 独立制御することで安定した制御を実現している。

光センサとして, 可視光領域と赤外領域の 2 つの光センサを備えている。可視光センサは可視光域の照度を検出し, マイコンの A/D 変換器を通して 10bit の分解能で検出できる。赤外線センサは, 赤外線領域の照度を検出し, マイコンの A/D 変換器を通して検出できるほか, 赤外線センサとして入力端子からデジタル入力として検出できる。そのことから, プログラムを介して赤外線リモコンなどの赤外線情報を取得できる。

光出力用の LED として, RGB フルカラー LED 及び赤外線 LED を備えている。RGB フルカラー LED は, 各色を独立した 8bit PWM 制御しており, 24bit の色を出力可能で, 実運用として 10bit 程度の色分解能を有する。赤外線 LED は, 8bit PWM 制御及びデジタル制御が可能である。赤外光の照度制御と共に, デジタル制御することで赤外線リモコンとして使用できる。

5. 応用システム

このデバイスを用いていくつかの応用システムを構築している。

(1) 電飾ダンス服¹³⁾

筆者らの研究グループで本デバイスを服に縫い込んだ電飾服を制作した。一つの服に約 300 個のデバイスを取り付けたものを二着用いて、着用者が音楽に合わせてダンスを踊るようなシステムを作っている。グローバル線には同期用のカウント信号を流し、ローカル線は用いていない。各ノードには LED の発光パターンのデータを格納し、カウント信号を見ながら LED を発光させる。万一カウント信号を受け損なった場合でも内部のタイマを用いて発光させる。こうすることで激しいダンスを踊りながらも安定的に LED を光らせることができる。現時点ではローカル通信は用いていないが、ローカル通信を用いれば加速度センサの値を使って加速度の生じた場所のあたりをローカルにパターンを変えるなど、動的なローカル処理が行えるようになるものと考えられ、より効果的な電飾ダンスの演出が行えるようになるものと考えられる。

(2) モバイルエージェント¹⁴⁾

グローバルグリッド上で動作するモバイルエージェントシステムを実装している。各デバイス上でインタープリタを動作させ、プログラムを受信するとそれを実行するような動作をさせた。ローカル通信のみを用いてエージェントが移動するため高度なローカルコンピューティングが可能になる。本稿で述べたさまざまなアルゴリズム(大規模リンク欠損検出など)についても、このモバイルエージェントのプログラムを用いるとより簡単に実現できる。3.2 節で述べた同時個別通信の枠組みを拡張して、さまざまなコマンドを考えるとモバイルエージェントの各コマンドに相当する機能が同時に多数実行できるようになるかもしれない。このような枠組みについても今後検討してゆきたい。

6. おわりに

ローカルトポロジとグローバルトポロジの関係が比較的自明な格子状の有線ネットワークについて、ローカル通信とグローバル通信を用いたコンピューティング手法について述べた。ソフトウェアの実装はまだ不完全なため早急に完成させたい。グローバルに指示を出し、ローカルにいろいろ調べたり、問題を解決したりしていくことはスケラビリティの観点から非常に有効性が高い。データを集約することや同時個別通信に関してはまだまだ考察が不十分であるが、ツリー構造などを使って面白いやり方を今後考えていきたい。また、格子状の枠組みは一般トポロジにも拡大可能であるものと考えられる。今後さらに適用範囲を広げていきたい。

謝辞 本研究は科研費基盤(A) 23240010 (研究代表 塚本昌彦) ならびに若手(B) 23700083 (研究代表 藤田直生) によるものである。ここに記して謝意を表す。

参考文献

- 1) 塚本, "ウェアラブル・ユビキタスコンピューティング - 超小型コンピュータと人, 物, 実世界のシンビオシス," 情報処理学会誌, Vol.47, No.8 (Aug. 2006).
- 2) 木下, 藤田, 柳沢, 寺田, 塚本, "分散制御された LED マトリックスを用いた電飾アート制御プラットフォーム," 情報処理学会研究報告(2009-EC-12), Vol. 2009, No. 26, pp. 65-70 (Mar. 2009).
- 3) M. Nakata, K. Kodama, N. Fujita, Yoshinari TAKEGAWA, T. Terada., M. Tsukamoto, S. Hosomi, and S. Nishio, "Design and Implementation of a Ubiquitous Optical Device Controlled with a Projector," Proc. of Int'l Conference on Advances in Mobile Computing & Multimedia (MoMM2008), pp. 130-135 (Nov. 2008).
- 4) K. Kodama, N. Fujita, Y. Yanagisawa, T. Terada, and M. Tsukamoto, "A Rule-based Acceleration Data Processing Engine for Small Sensor Node," Proc. of the 3rd Int'l Workshop on Middleware for Sensor Networks (MidSens2008), pp. 1-6 (Dec. 2008).
- 5) N. Fujita, T. Yoshihisa, and M. Tsukamoto, "A Sensing System for Data Collection with Broadcasting Predicted Data," Proc. of 2008 Fourth Int'l Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP2008), pp. 465-470 (Dec. 2008).
- 6) C. W. Kim, M. Kawatani, M. Tsukamoto, and N. Fujita, "Wireless Sensor Node Development for Bridge Condition Assessment," Proc. the 3rd Int'l Conference Smart Materials, Structures and Systems, Transactions on Advances in Science and Technology, Vol. 56, pp. 573-578 (June 2008).
- 7) S. Hosomi, M. Tsukamoto, and S. Nishio, "Cooperation of LED Control Chips in Ubiquitous Environment," Proc. of IASTED Int'l Conf. on Circuits, Signals, and Systems (CSS 2007), pp. 249-254 (July 2007).
- 8) K. Whitehouse, C. Sharp, E. Brewer, and D. Culler, "Hood: a Neighborhood Abstraction for Sensor Networks," Proc. of the 2nd Int. Conf. on Mobile Systems, Applications, and Services (MOBISYS) (2004).
- 9) R. Gummadi, O. Gnawali, and R. Govindan, "Macroprogramming Wireless Sensor Networks using Kairos," Proc. of the 1st Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)(2005).
- 10) L. Mottola and G. P. Picco, "Programming Wireless Sensor Networks with Logical Neighborhoods," Proc. of the 1st Int. Conf. on Integrated Internet Ad hoc and Sensor Networks (InterSense)(2006).
- 11) R. Newton and M. Welsh, "Region Streams: Functional Macroprogramming for Sensor Networks," Proc of the 1st Int. Workshop on Data Management for Sensor Networks (DMSN)(2004).
- 12) A. Pathak, L. Mottola, A. Bakshi, G. P. Picco, and V. K. Prasanna, "A Compilation Framework for Macroprogramming Networked Sensors," Proc. of the 3rd Int. Conf. on Distributed Computing on Sensor Systems (DCOSS)(2007).
- 13) M. Fujimoto, N. Fujita, T. Terada, and M. Tsukamoto, "Lighting Choreographer: an LED Control System for Dance Performances," Video Paper of Ubicomp 2011, to appear.
- 14) 國本, 藤田, 佐野, 寺田, 塚本, "格子状に接続されたユビキタスコンピュータ群のモバイルエージェントを用いた制御手法," 情報処理学会マルチメディア, 分散, 協調とモバイル (DICOMO2011)シンポジウム論文集 (July 2011).