

M-85 ユビキタスコンピューティングのための入出力制御デバイスのPC統合環境 A PC Integration Environment of I/O Control Devices for Ubiquitous Computing

岸野 泰恵*1 義久 智樹*1 寺田 努*2 塚本 昌彦*1
 Yasue Kishino Tomoki Yoshihisa Tsutomu Terada Masahiko Tsukamoto
 坂根 裕*3 早川 敬介*4 柏谷 篤*4 西尾 章治郎*1
 Yutaka Sakane Keisuke Hayakawa Atsushi Kashitani Shojiro Nishio

1. はじめに

近年、いつでもどこでもコンピュータを使用できるユビキタスコンピューティングに対する注目が高まっている。筆者らの研究グループでは、ユビキタスコンピューティング環境の実現にむけて、ECAルール形式で動作を記述する入出力制御デバイスを用いたユビキタスコンピューティングを提案している[1]。提案する入出力制御デバイスは、小型で、さまざまな機器やセンサなどと連携させることを想定している。本稿では、この入出力制御デバイスをユビキタスチップと呼ぶ。

ユビキタスチップは ECA ルールを用いて動作を記述するが[2]、ルールはビット列で表現されているため直感的に内容を把握できず、ルールの作成やデバッグが困難であった。そのため、多くのルールが連動して動作するようなアプリケーションを構築しにくいという問題があった。そこで本稿では、ルールを PC 上で直感的に作成でき、作成したルールを簡単にユビキタスチップに書き込める開発環境について述べる。

さらに本稿では、PC 上で動作するユビキタスチップのエミュレータを用いて PC とユビキタスチップの連携を可能にする PC 統合環境について述べる。PC 統合環境を用いることで、ユビキタスチップを PC や PDA、携帯電話などと連携させて動作させ、これらの機器を通じてネットワーク上からユビキタスチップを制御するようなサービスの提供が可能となる。

2. ユビキタスチップの概要

提案するユビキタスチップは 5 ポートの入力ポートと 12 ポートの出力ポート、2 ポートのシリアル通信ポート、タイマ、ユビキタスチップ間のメッセージ送受信機能をもつ[3]。ユビキタスチップの動作を記述したルールは 128 バイトの容量のメモリに格納し、ルールの消去や追加はシリアル通信でコマンドを送受信して行う。コマンドには、ルールを追加する ADD_ECA、格納されているルールを全て削除する CLEAR_ECA、アドレスを指定してメモリの内容の送信を要求する REQUEST_ECA がある。さらに、コマンドを発生させるルールも記述できるため、ルールによって接続している他のユビキタスチップにルールを追加することもできる[4]。

3. ECA ルール開発環境

本稿では、ユビキタスチップに格納する ECA ルールの作成を容易にし、アプリケーションの開発を支援する環境として、ECA ルールコンパイラと ECA ルールライターについて述べる。

*1: 大阪大学大学院情報科学研究科, Graduate School of Information Science and Technology, Osaka University

*2: 大阪大学サイバーメディアセンター, Cybermedia Center, Osaka University

*3: 大阪大学大学院工学研究科, Graduate School of Engineering, Osaka University (現在, 静岡大学情報学部)

*4: NEC インターネットシステム研究所, Internet Systems Research Laboratories, NEC Corp.

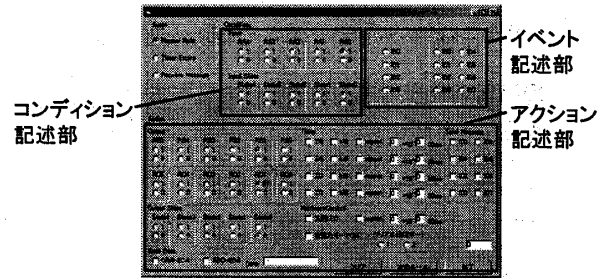


図 1: ECA ルールコンパイラの入力画面

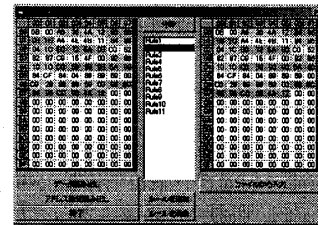


図 2: ECA ルールライタの制御画面

3.1 ECA ルールコンパイラ

ECA ルールコンパイラは、ユーザがユビキタスチップ上で動作させたいルールを入力すると、単純なバイナリの形式で記述されたルールのファイルを出力するアプリケーションである。

ECA ルールはイベント、コンディション、アクションの 3 つを一組にして動作を記述する言語であり、図 1 に示す入力画面では、イベント記述部でメッセージの受信、タイマの発火といった条件を、コンディション記述部で入力状態と内部状態を、アクション記述部で出力やタイマの設定、メッセージの送信などの動作をマウス操作で指定する。最後にルール作成ボタンを押すと、入力した内容がルールに変換される。ECA ルールコンパイラを用いると、ルールを作成する際にフォーマットを確認する、ビット列を 16 進数に換算するといった面倒な作業が不要となり、容易にルールを作成できる。

3.2 ECA ルールライター

ECA ルールライターでは、シリアル通信を用いてユビキタスチップへコマンドを送信し、ルールの消去や追加、書き込みが行える。図 2 に ECA ルールライタの制御画面を示す。

ルールを書き込むときには、コンパイラで出力されたファイルを読み込み、ルールを分析して個々のルールに分ける。1 つのルールのみを追加するときは選択されたルールを ADD_ECA コマンドを利用して送信し、全てのルールを書き込むときは順に全てのルールを送信する。ルールを削除するときには、CLEAR_ECA コマンドを送信する。本プログラムによって PC とシリアル通信で接続したユビキタスチップのルールを自

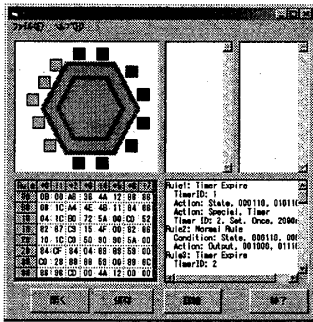


図 3: ユビキタスチップエミュレータの画面

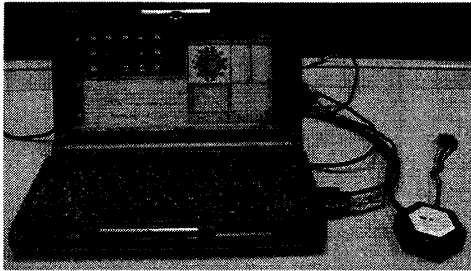


図 4: ユビキタスチップと PC の接続例

由に制御できる。

また、このプログラムはユビキタスチップに格納されたルールの読み取り機能をもっている。全てのアドレスについて REQUEST_ECA を繰り返すことでルールを読み取り、読み取ったルールを分析して、別のユビキタスチップに書き込める。

4. PC 統合環境

ユビキタスチップはシリアル通信を用いて、メッセージやコマンドを送受信できる。この機能を用いて、ユビキタスチップに接続したセンサーから得られる情報を PC で利用したり、ユビキタスチップに格納されているルールを PC で制御するなど、PC とユビキタスチップを連携させることで多様なサービスが提供できる。そこで本研究では、PC とユビキタスチップを統合利用するために、PC 上にユビキタスチップエミュレータを実装した。

図 3 にエミュレータの画面を示す。エミュレータも、ユビキタスチップで用いているものと同じフォーマットのルールで動作を記述できる。エミュレータにおけるユビキタスチップの入力ポートの状態はマウスでクリックして制御する。また、出力ポートの状態は色の明暗で表現している。図 4 にユビキタスチップを PC と接続して、エミュレータと通信している例を示す。

ユビキタスチップエミュレータは PC 上で動作するため、ルールを記憶するメモリの大きさに制限がない。この特徴を利用し、ユビキタスチップと同じように動作させだけでなく、他のユビキタスチップのためのルールをメモリに格納し、接続してきたユビキタスチップへルールを配信するといった利用法も可能である。

5. 考察

ECA ルールコンパイラ

実装した ECA ルールコンパイラを使用したところ、図 1 の画面から分かるように、マウスで多くのボタンをクリックすることでルールを作成するため、同じ操作を何度も繰り返さなければならなかったり、クリックする場所を間違えるという問

題があった。大規模なアプリケーション開発を支援するためには、自動的にピンの配置を決定するようなプログラムや、ユビキタスチップ用のサービス記述言語を導入する必要がある。

ユビキタスチップは直接電圧の高低を操作できるので、誤ったルールで接続しているデバイスを故障させる可能性がある。今後は、作成したルールが意図した通りに動作するのか検証し、その安全性を確認できるようなデバッガが必要となる。さらに、複数のユビキタスチップが連携するようなアプリケーションも検証できるようにしたいと考えている。

応用

ユビキタスチップと PC や携帯端末が連携することで、必要に応じてネットワーク上からルールをダウンロードしてユビキタスチップに格納することや、複数のユビキタスチップからセンサの情報を集めて PC で管理すること、PC でそれらの情報を集約した結果をもとに新たにイベントを発生させることが可能になる。

例えば、将来的には次のような応用を考えている。

- 朝、家を出るのが少し遅かったが、携帯端末が自動的に検索した結果、最寄駅まで早足で行けば電車に間に合うことが分かった。ユビキタスチップは振動モータを一定のテンポで振動させる。このテンポで歩けば、電車に間に合う。
- 改札を通過すると、ユビキタスチップには車内用のルールが格納される。車内では、空調が強すぎるため、携帯端末を操作し、寒すぎるというボタンを押す。同じ車両に乗っている人のユビキタスチップ同士が通信し、寒すぎると思う人が一定の割合を越えると、車両についているユビキタスチップが自動で空調を弱める仕組みになっている。

これらの応用を可能にするためには、PC や携帯端末とユビキタスチップが連携するためのより高度なプログラム、ユビキタスチップの無線通信への対応などが必要である。

6. 最後に

本稿では、これまでに提案したユビキタスチップに搭載する ECA ルールの開発と、PC 連携をサポートする環境について述べた。

今後、より高度なアプリケーション開発・デバッグ環境を実現する。具体的には、1 台の PC の上で大量のユビキタスチップを相互に接続して動作を確認できるシミュレータの作成、PC 上でユビキタスチップエミュレータへ入力できる仮想センサの実装などを行う予定である。

参考文献

- [1] 塚本ほか: ユビキタスコンピューティングを実現するためのルールに基づく入出力制御デバイス, FIT2002 論文集 (2002 掲載予定)。
- [2] 寺田ほか: ユビキタスコンピューティングのための入出力制御デバイスの動作記述言語, FIT2002 論文集 (2002 掲載予定)。
- [3] 早川ほか: ユビキタスコンピューティングのための入出力デバイスのハードウェアアーキテクチャ, FIT2002 論文集 (2002 掲載予定)。
- [4] 義久ほか: ユビキタスコンピューティングのための入出力制御デバイスのソフトウェアアーキテクチャ, FIT2002 論文集 (2002 掲載予定)。