

ユビキタス環境における LED 明滅の連携動作記述方式

細見 心一[†] 塚本 昌彦[‡] 西尾 章治郎[†]

[†] 大阪大学大学院情報科学研究科

[‡] 神戸大学工学部電気電子工学科

近年、光る電子装置を使ったアクセサリやアイテムが注目を集めている。これらの電子装置は小型入出力制御デバイスを用いることで、装置同士の複雑な連携動作が可能となる。しかし、装置同士の全体的な動作を記述するグローバルなプログラミング方式はこれまでにない。そこで、本稿では、小型入出力制御デバイスの LED 明滅を制御できるグローバルなプログラミング方式を提案する。これにより、小型入出力制御デバイスの群制御や分散処理を簡単に記述することができる。提案手法を小型入出力デバイスに実装し、有効性について述べる。

A Method for Describing Cooperating Behavior on Blinking LEDs in Ubiquitous Environment

Shinichi Hosomi[†] Masahiko Tsukamoto[‡] Shojiro Nishio[†]

[†] Graduate School of Information Science and Technology, Osaka University

[‡] Faculty of Engineering, Kobe University

Recently, accessories using luminous electronic equipments have attracted attention. An I/O(input/output) control device enables these items to realize cooperating behavior among them. However, there have been no way to globally describe the behavior of multiple computers. In this paper, we propose a description method for cooperating behavior among blinking items. Our proposed method realizes global programming by controlling blinking patterns. Moreover, we discuss on the validity of this system on our implemented systems.

1 はじめに

近年、コンピュータや半導体、センサなどの小型化により、ユビキタスコンピューティング環境の実現に向けての多くの研究開発が行われている。これまで、IC タグや埋め込みコンピューティング技術が広く使われているが、これらは主としてあらかじめ決められた動作を行うものであり、次のような高度な活用のためには、より高度なメカニズムが必要となる。

- 汎用性：さまざまな用途に利用できること
- 柔軟性：さまざまな環境で利用できること
- 有機性：別の機器と連携して利用できること

従来から、高度な柔軟性や適応性を有するコンピューティングシステムについては、分散人工知能や分散知識処理、エージェントなどの分野で盛んに研究が行われている。これらの研究成果の多くは、高度な計算をコンピュータに課すものであり、ハードウェアに制約があるユビキタスコンピューティングへの適応は困難である。

一方、ハードウェアの制約を考慮したユビキタスデバイスに関する研究も数多く行われている。MOTE は無線センサネットワークのためのプラットフォームである [6]。Smart-Its はさまざまなものに埋め込むための小型コンピュータである [2]。

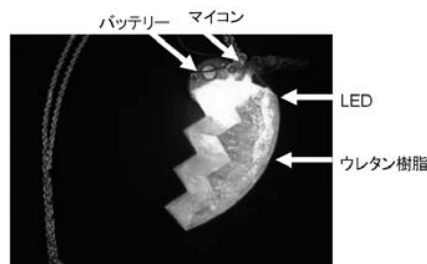


図 1: Bloom アクセサリ

AhroD はイベント駆動型ルールを交換することで動的に動作を変更できる入出力制御デバイスである [4]。マイクロ Ja-Net は、モバイルエージェントの技術を用いて、単純なマイクロコードを移動実行できるエンジンを搭載する [7]。特定のハードウェア専用のアプリケーション開発環境としては、センサネットワーク端末向け動作環境である TinyOS [9]、Ahrod のアプリケーション開発環境 [8]、LEGO 社の MINDSTORMS [5] や、バンダイ社の ROBOT WORKS [1] などがある。

これらのユビキタスデバイスを、近年増加傾向にあるイルミネーション機能付き電子ガジェットに組み込むことで、動的な光の制御が可能になる。イルミネーション機能付き電子ガジェットは、携帯電話、



図 2: ウェアラブルコンピュータショウ（大阪）でのイベントの様子



図 3: WMTE 2005（徳島）でのイベントの様子

光るバッチ、光るストラップ、光るアクセサリなどがある。これまで、筆者らのグループでは、電飾アクセサリや電飾アイテムを提案している [11]。電飾アクセサリは、高輝度の 3 色の LED をウレタン樹脂に埋め込み、明滅パターンをマイコンで制御する（図 1）。従来のアクセサリと比較すると、色や光を動的に変更できるため、ユーザや制作者の意図をより柔軟に表現できるという利点を持つ。また、電飾アクセサリや電飾服が、ウェアラブルコンピュータショウ in KANSAI 併催：IEEE International Symposium on Wearable Computers (ISWC 2005) と IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2005) のパンケットで披露された。前者は、2005 年 10 月 19 日と 20 日に大阪産業創造館にて開催されたファッションショーで、電飾アイテムを服に取り付けた作品が多数披露された（図 2）。後者は、2005 年 11 月 29 日に徳島にて開催され、電飾アイテムを使った阿波踊りが披露された（図 3）。高輝度 LED を取り付けた光る小手や光る爪、炎のゆらぎを再現した光る提灯（図 4）を利用し、光るタイミングを無線で通信し同期をとる試みが行われた。このように、電飾アイテムは学術的にも注目を集めている。他に、2006 年 7 月 29 日と 8 月 5 日に摩耶山掬星台にて開催されたイベント「第 2 回 星と光の祭典」にて、ドレス、風船、十字架、ハンカチ、手袋についた LED を連携動作させた（図 5）。本稿では、小型入出力制御デバイスの LED 明滅を同期させることを光の群制御と呼ぶことにする。



図 4: ゆらぎ提灯



図 5: 摩耶山でのイベントの様子

従来のユビキタスデバイスで光の群制御を行う場合、それぞれ次の問題がある。MOTE や Smart-Its はファームウェアをあらかじめ書き込んで使用するため、動的な動作変更ができない。Ahrod やマイクロ Ja-Net は動的なネットワークの変化に対応することができるが、全体的な動作をプログラミングする場合はその記述が複雑になる。TinyOS、LEGO 社の MINDSTORMS、バンダイ社の ROBOT WORKS では複数のデバイスの連携が考慮されていない。Ahrod のアプリケーション開発環境では、デバイスの全体的な動作をプログラミングすることはできない。

これらの問題を解決するためには、デバイスの全体的な動作を記述するグローバルなプログラミング方式が必要となる。特に、次の要求が考えられる。

- **デザイン性**: ユーザが意図した光の群制御を実現するために、デザインしやすいプログラミング方式が求められる。
- **拡張性**: 多様な形態の光の群制御を実現するため、プログラムを組み合わせることができる。
- **機能性**: 省メモリで低速なデバイスに実装できる。

しかし、これらの要求をすべて満たすプログラミング方式は確立されていない。そこで、本研究では、小型入出力制御デバイスの LED 明滅に着目し、光の群制御をグローバルに記述できるプログラミング



図 6: 電飾アクセサリの連携

方式を提案する。提案した手法では、PC 上の編集アプリケーションを利用し、小型入出力デバイスの群制御を実現する。

以下、2 章で提案手法であるインクリメンタルプログラミング方式の概要について述べ、3 章では、提案手法の詳細について述べる。また、4 章で本手法の実装について説明し、5 章で本手法の考察を行う。最後に 6 章でまとめを行う。

2 インクリメンタルプログラミング方式

本章では、本研究で想定するユビキタス環境における LED 明滅の群制御について説明し、ユビキタス環境における LED 明滅の群制御に求められる要求事項を明らかにする。最後に提案手法を説明する。

2.1 ユビキタス環境における LED 明滅の群制御

本研究では、図 6 のように、無線機能の付いた電飾アクセサリがシーケンシャルに配置され、ユーザがその位置を把握可能な状態を想定する。左から順にデバイス ID を 1, 2, 3, 4 … とする。電飾アクセサリの群制御は、次の形態が考えられる。

- 一気に光の色が変わる。
- 光が流れるように変化する。
- 光がウェーブするように変化する。

2.2 要求事項

上記のような電飾アクセサリの群制御を記述するグローバルなプログラミング方式には 1 章で述べたデザイン性、拡張性、機能性の要求に対して、以下の性質が必要となる。

- **デザイン性**: ユーザが意図した群制御を実現するために、デザインしやすいプログラミング方式が求められる。そのために、簡単な記述で群制御や分散処理を動的に実行するラピッドプロトタイピングができる。
- **拡張性**: 多様な群制御を実現するため、プログラムを組み合わせることができる。

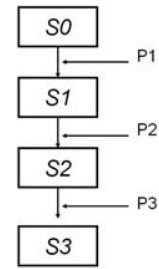


図 7: インクリメンタルプログラミング

- **機能性**: 小型入出力制御デバイスは、省メモリで低速なマイコンで動作する。そのため、マイコンの容量やメッセージングを考慮する必要がある。

2.3 提案手法の概要

本研究では、以上の要求事項を満たすインクリメンタルプログラミング方式を提案する。インクリメンタルプログラミング方式とは、追加するプログラムを P 、プログラムによる作用全体を S としたとき、図 7 のように、 S を逐次変更する方式と定義され、次のような特徴を持つ。

● デザイン工程への適応

インクリメンタルプログラミング方式は、プログラムを追加し逐次実行していくという特性から、以下の 2 点でデザインに適していると考える。

(1) 追加の繰り返し

デザインとは、描くなど、身体的な行為を繰り返す中で生みだされていく知的な創造過程である [3]。インクリメンタルプログラミング方式のプログラムを追加していく点が、デザインにおける身体的な行為の繰り返しにあたる。

(2) 思考プロセスの保存

デザインは、一般社会と共有できる問題を設定し、それを解決していくプランニングであり、そのプロセスは誰もが理解し進めることができる [3]。インクリメンタルプログラミング方式は、追加したプログラムの履歴を残すことで、思考プロセスを他のユーザと共有することができる。

● 拡張性

インクリメンタルプログラミング方式は、プログラムを無限に追加していくことができる。これにより、多様な形態の光の群制御を実現する。

表 1: プログラムの一覧

プログラム	処理内容
write	小型入出力制御デバイスに明滅パターンを書き込む
erase	小型入出力制御デバイスの明滅パターンを消去
light	小型入出力制御デバイスの明滅パターンを再生
stop	小型入出力制御デバイスの明滅パターンの再生を停止
ledx = dutyrate(y) in t sec	ポート x で、明るさ y の光を t 秒間生成
ledx = lightup / lightdown between a and b in t sec	ポート x で、t 秒間かけて明るさ a から b までの範囲で、 徐々に明るく / 徐々に暗くしていく。between は省略可能で、 省略したときは a=0, b=127 となる。
ledx = blink(s) a in t sec	ポート x で、明るさ a の、点滅を t 秒間表示する。 between は省略可能。
from x1 to x2 dark	x1 秒から x2 秒までの明滅パターンを暗く (発光強度 $\times 3/4$)
from x1 to x2 bright	x1 秒から x2 秒までの明滅パターンを明るく (発光強度 $\times 5/4$)
from x1 to x2 slow	x1 秒から x2 秒までの明滅パターンを波形を変えずに 2 倍に伸ばす
from x1 to x2 fast	x1 秒から x2 秒までの明滅パターンを波形を変えずに 1/2 に縮める
from x1 to x2 long	x1 秒から x2 秒までの明滅パターンを 2 回繰り返す
from x1 to x2 short	x1 秒から x2 秒までの明滅パターンを 1/2 に省略する
from x1 to x2 skip	x1 秒から x2 秒までの明滅パターンを削除する
from x1 to x2 shift	x1 秒から x2 秒まで、明滅パターンをずらす
from x1 to x2 brighter	デバイスの明滅パターン x1 秒から x2 秒までを左からシーケンシャルに明るく デバイス ID (ID):1 発光強度 $\times 5/4$, ID:2 発光強度 $\times 25/16$, ID:3 ...
from x1 to x2 darker	デバイスを明滅パターン x1 秒から x2 秒までを左からシーケンシャルに暗く デバイス ID (ID):1 発光強度 $\times 3/4$, ID:2 発光強度 $\times 9/16$, ...
from x1 to x2 faster	デバイスを明滅パターン x1 秒から x2 秒までを左からシーケンシャルに速く デバイス ID (ID):1 速度 $\times 2$, ID:2 速度 $\times 4$, ...
from x1 to x2 slower	デバイスを明滅パターン x1 秒から x2 秒までを左からシーケンシャルに遅く デバイス ID (ID):1 速度 $\times 1/2$, ID:2 速度 $\times 1/4$, ...

from 0 to 1 dark
from 0 to 1 slow
from 0 to 3 long

図 8: 明滅パターンの編集プログラム例

● メタプログラミング

小型入出力制御デバイスは、省メモリで低速なマイコンで動作する。そのため、マイコンの容量やメッセージングを考慮した設計が必要である。インクリメンタルプログラミング方式は追加されるプログラムをメタプログラムとして処理し、小型入出力制御デバイスで分散処理を行う。これにより、使用するメモリ容量と通信パケットを少なくすることができる。

3 インクリメンタルプログラミング方式の詳細

本章では、本研究で提案するインクリメンタルプログラミング方式の詳細について述べる。

3.1 コマンドの設計

LED の明滅パターンを生成するために、表 1 のコマンドを用いる。PC 上の編集ソフトウェアを利用しコマンドの入力を行い、小型入出力制御デバイスが再生する明滅パターンを動的に変更させる。図 8 に明滅パターンの編集プログラム使用例を、図 9 に編集される明滅パターンの例を示す。デバイスごと

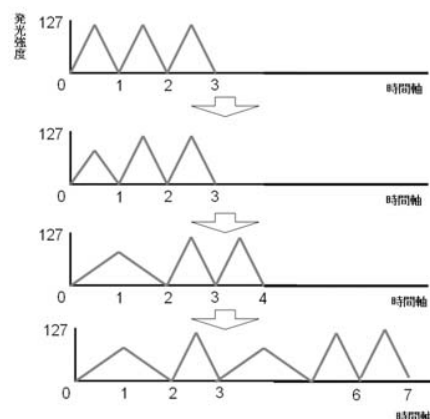


図 9: 明滅パターンの編集

の明滅パターンを変えることで、光が流れるといった光の群制御を実現する。例えば、デバイスが 4 つ存在し、共通の明滅パターンを再生しているとす。ユーザが from 0 to 10 brighter と入力した場合、ID:1 のデバイスは bright を 1 回実行し、ID:2 のデバイスは bright を 2 回実行し、ID:3 のデバイスは bright を 3 回実行し、ID:4 のデバイスは bright を 4 回実行する (図 10)。ユーザが from 0 to 1 slower と入力した場合、ID:1 のデバイスは slow を 1 回実行し、ID:2 のデバイスは slow を 2 回実行し、

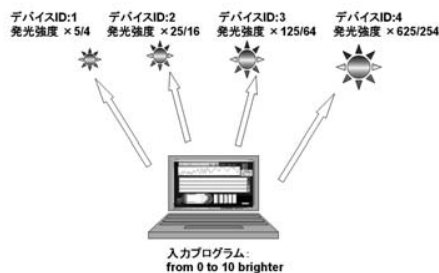


図 10: from 0 to 10 brighter と入力した場合におけるデバイスでの処理

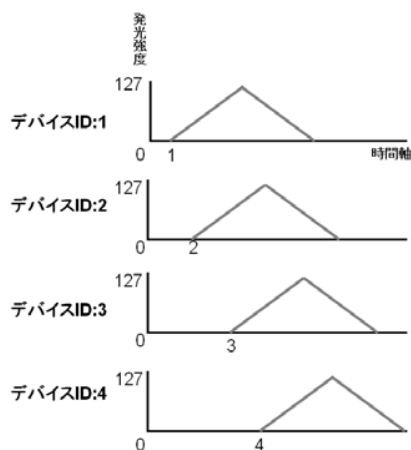


図 11: from 0 to 1 slower と入力した場合における明滅パターン

ID:3 のデバイスは slow を 3 回実行し、ID:4 のデバイスは slow を 4 回実行する。その結果、図 11 に示すようなウェーブを、4 つのデバイスで実現することができる。

3.2 メタプログラミング

省メモリで低速なマイコンに対応するため、PC から送信される明滅パターンの編集に関するプログラムをメタプログラムとして、小型入出力制御デバイス内でプログラムを生成する。PC から送信される明滅パターンの編集に関するプログラムは、dark, bright, slow, fast, long, darker, brighter, slower, faster, longer である。小型入出力制御デバイス内で生成されるプログラムは、LED 明滅データの更新・挿入・削除を行うプログラムである。メタプログラムから生成されるプログラムをバイトコード型プログラムと呼ぶことにする。バイトコード型プログラムは、バイトコードで保存する。PC

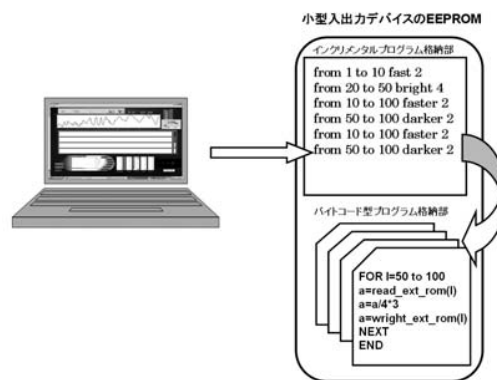


図 12: コマンドの保存とプログラムの作成

から送られてくるコマンドは、EEPROM に履歴を保存していくが、バイトコード型プログラムは同じ番地に逐次上書きしていく (図 12)。これにより、プログラムの履歴を省メモリに残すことができる。PC からの通信パケット数を減らすことができる。メタプログラムから生成されるバイトコード型プログラムを図 13、図 14 に示す。read_ext_prom (I), write_ext_prom (J) は独自の関数で、EEPROM の I 番地のデータを読み込み、J 番地に保存する。last は明滅データの最終番地を表す。

4 実装

本システムは、PC と小型入出力制御デバイスで構成される。PC 上の編集アプリケーションを利用し、小型入出力制御デバイスの群制御を行う (図 15)。

4.1 編集アプリケーション

PC 上では、図 16 の編集アプリケーションを用いてコマンドの入力を行う。コマンドの入力以外に、明滅パターンのマウス描画エディタを用意した。マウス描画エディタでは、横軸を時間、縦軸を発光強度とする明滅パターンをマウスで描くことができる。

4.2 小型入出力制御デバイス

小型入出力制御デバイス (図 17) は、12 ポートの出力ポートと 2 ポートのシリアル通信ポート、タイマ、小型入出力制御デバイス間のメッセージ送受信機能を持つ。CadSoft Computer 社のプリントパターン作成ソフト EAGLE を使用し、独自に設計開発を行った。基板の大きさは、40 × 60 × 25 (mm) である。ボタン電池で駆動し、基板の裏に電池ボックスがある。基板上のコネクタに LED モジュールを接続する。フルカラー LED を最大 4 個接続可能である。小型入出力制御デバイスはコア部と通信部に別れ、コア部ではマイクロチップテクノ

```

// from x1 to x2 dark x
//から生成されるプログラム
FOR I=x1 TO x2
a=read_ext_rom(I)
a=a/4*3
a=wright_ext_rom(I)
NEXT
END

// from x1 to x2 bright x
//から生成されるプログラム
FOR I=x1 TO x2
a=read_ext_rom(I)
a=a/4*5
a=wright_ext_rom(I)
NEXT
END

// from x1 to x2 skip
//から生成されるプログラム
J=x1
FOR I=x2 TO last
a=read_ext_rom(I)
a=wright_ext_rom(J)
J=J+1
NEXT
END

// from x1 to x2 short
//から生成されるプログラム
J=x1
FOR I=((x1+last)/2) TO last
a=read_ext_rom(I)
a=wright_ext_rom(J)
J=J+1
NEXT
END

// from x1 to x2 long x
//から生成されるプログラム
J=last
K=last+((x1+x2)/2)
FOR I=x1 TO last
a=read_ext_rom(J)
a=wright_ext_rom(K)
J=J-1
K=K-1
NEXT
END

```

図 13: dark, bright, skip, short, long から生成されるバイトコード型プログラム

ロジ社の PIC16F876（プログラムメモリ：8K バイト，RAM：368 バイト，EEPROM：256 バイト）を 20MHz で動作させて用いている。補助記憶装置として、32K バイトの EEPROM を用いている。通信部は無線通信ユニット（有限会社ケイツー電子工業，WCU-242）と周辺回路で構成される。シリアル通信ポートを介して、通信速度 9600bps，データ長 1 バイトの RS232C 通信を行う。

LED の明滅には、アナログ波形のパターンを再生するために、PWM（Pulse Width Modulation）制御を用いた。人間の目の周波数応答が約 60Hz であることから、PWM 制御の周期を 13 ミリ秒とし、duty 比の分解能を 7 ビットの 128 段階とする。この 13 ミリ秒ごとの 0～127 の発光強度を 1 バイト

```

// from x1 to x2 fast
//から生成されるプログラム
J=x1
FOR I=x1 TO ((x1+x2)/2)
a=read_ext_rom(J)
a=wright_ext_rom(I)
J=J+1
NEXT
FOR J=x2 TO last
a=read_ext_rom(J)
a=wright_ext_rom(I)
I=I+1
NEXT
END

// from x1 TO x2 slow
//から生成されるプログラム
J=last
K=last+((x1+x2)/2)
FOR I=x2 TO last
a=read_ext_rom(J)
a=wright_ext_rom(K)
J=J-1
K=K-1
NEXT
J=x2
FOR I=x1 TO x2
a=read_ext_rom(J)
a=wright_ext_rom(K)
K=K-1
a=read_ext_rom(J)
a=wright_ext_rom(K)
J=J-1
K=K-1
NEXT
END

```

図 14: fast, slow から生成されるバイトコード型プログラム

の明滅データとして補助記憶装置に格納する。

小型入出力制御デバイスは、専用のライターでファームウェアをあらかじめ書き込み、明滅データを補助記憶装置に格納して動作する。小型入出力デバイスのシステム構成を図 18 に示す。メタプログラム処理では、メタプログラムからバイトコード型プログラムを生成する。データ書き換え処理では、生成されたバイトコード型プログラムの処理に従い、補助記憶装置に格納されている明滅データを編集する。

4.3 通信プロトコル

無線フォーマットは、1 パケット 1 バイトとする。上位 3 ビットをコマンド ID，下位 5 ビットをデータとする。表 2 にコマンド ID とデータ内容を示す。

4.4 バイトコード型プログラムのフォーマット

バイトコード型プログラムは、バイトコードで保存される。以下に、バイトコード型プログラムのフォーマットを示す。



図 15: 全体の機器構成

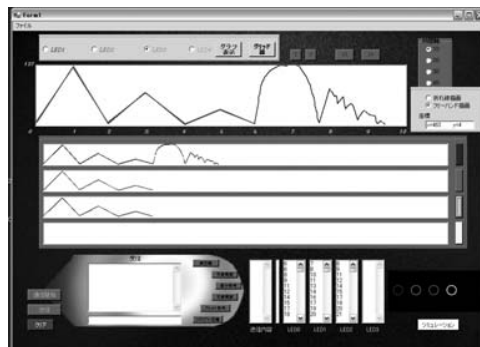


図 16: 編集アプリケーション

FOR I=y1 to y2

5 バイト使用する。1 バイト目の上位 4 ビットをコマンド ID, 下位 4 ビットを変数番号とする。y1, y2 とともに 2 バイト使用する。変数番号は, 16 個あらかじめ用意した long 型の変数を指定する。

NEXT I

1 バイト使用する。上位 4 ビットをコマンド ID, 下位 4 ビットを変数番号とする。

I=y3 or I=I+y3 or I=I-y3

3 バイト使用する。1 バイト目の上位 4 ビットをコマンド ID, 下位 4 ビットを変数番号とする。y3 で 2 バイト使用する。

a=read_ext_rom (I)
a=write_ext_rom (J)

2 バイト使用する。1 バイト目の上位 4 ビットをコマンド ID, 5 ビット目をフラグとする。2 バイト目の上位 4 ビットを read の変数番号, 下位 4 ビットを write の変数番号とする。フラグが立っている場合は, 全部で 3 バイト使用する。3 バイト目の上

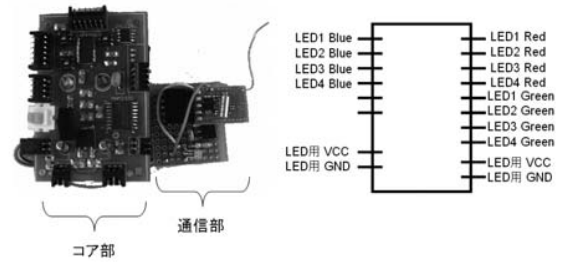


図 17: 小型入出力制御デバイス

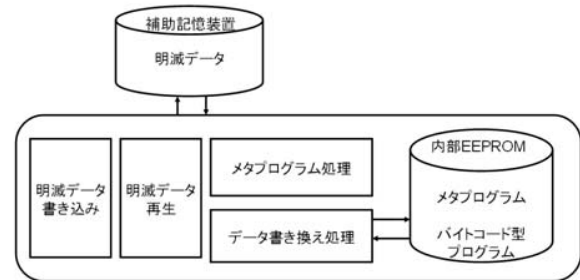


図 18: 小型入出力デバイスのシステム構成

位 4 ビットを分母, 下位 4 ビットを分子とする乗算を, read_ext_rom と write_ext_rom の間で行う。

5 考察

提案手法の有効性を示すため, 使用するメモリ量について比較を行った。PC から送信されるコマンドとバイトコード型プログラムで使用するメモリ量の比較した結果を表 3 に示す。PC から送信されるコマンドはバイトコード型プログラムと比較して, 省メモリで履歴の保存が可能となる。

6 まとめ

本稿は, イルミネーション付き電子ガジェットの群制御に着目し, 小型入出力制御デバイスの LED 明滅の群制御をグローバルに記述できるプログラミング方式を提案した。提案した手法では, PC 上の編集アプリケーションを利用し, 小型入出力デバイスの LED 明滅群制御を実現した。PC から送信されるコマンドをメタプログラムとすることで, 省メモリの小型入出力制御デバイスへの実装が可能になる。今後の課題として, 時相論理, 区間時相論理を考慮した記述方法について設計を行う。

謝辞 本研究の一部は, 文部科学省特定領域研究「情報爆発のための装着型入出力デバイスを用いた

表 2: 通信プロトコル

コマンド ID	データ内容
001	明滅データ編集用プログラムの コマンド ID
010	from x1 (上位 5 ビット)
011	from x1 (下位 5 ビット)
100	to x2 (上位 5 ビット)
101	to x2 (下位 5 ビット)

表 3: メモリ量の比較

命令	コマンド (バイト)	バイトコード型 (バイト)
dark	7	10
bright	7	10
short	7	15
fast	7	25
long	7	24
slow	7	43

情報操作方式」(18049058)によるものである。ここに記して謝意を表す。本研究を進めるにあたり、多大なご協力を頂いたチームつかもと、神戸ベンチャー研究会、上田安子服飾専門学校の関係者の方々に感謝の意を表す。

参考文献

- [1] バンダイ: ROBOT WORKS,
<http://www.roboken.channel.or.jp/borg/>.
- [2] Beigl, M. and Gellersen, H.: Smart-Its:An Embedded Platform for Smart Objects, *SmartObjectsConference(sOc)*, 2003
- [3] 奥出直人, 後藤武: 『デザイン言語』—感覚と論理を結ぶ思考法, 慶応義塾大学出版会
- [4] 早川敬介, 塚本昌彦, 寺田努, 義久智樹, 岸野泰恵, 柏谷篤, 坂根裕, 西尾章治郎: ユビキタスコンピューティングのためのルールに基づく入出力制御デバイス, ヒューマンインタフェース学会論文誌, Vol.5, No.3, pp.341–354, 2003.
- [5] LEGO: MINDSTORMS,
<http://mindstorms.lego.com/japan/products/>.
- [6] MICA. http://www.xbow.com/products/Wireless_Sensor_NetWorks.htm
- [7] Satoshi N., Takefumi O., Masahiko T. and Syojiro N.: Ja-Net on Grid for Adaptive Resource Distribution in Grid Environment, *Proc.ofInt'lConf.onActiveMediaTechnology(AMT2005)* , pp. 409–414, 2005.
- [8] Ryhei, S., Yasue K., Tsutomu T., Tomoki Y., Masahiko T. and Syojiro N.: An Application Development Environment for Rule-based I/O Control Devices, *Proc.ofInt'lSymposiumonUbiquitousIntelligenceandSmartWorlds(UISW2005)*, pp.121–130, 2005.
- [9] TinyOS,
<http://webs.cs.berkeley.edu/tos/index.html>.
- [10] Weiser, M.: The Computer for the 21st Century, *ScientificAmerican*, Vol.265, No.3, pp.94–104, 1991.
- [11] Yasue K., Hiroyuki F., Toshiyuki T., Asao S., Tomoki Y., Masahiko T., Tomoko I., Misuko O. and Syojiro N.: Bloom Accessory: Accessories Using LEDs with Remote Control, *Proc.ofIEEEInt'lSymposiumonWearableComputers(ISWC2004)* , pp.180–181, 2004.