社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－MBL－36 (50)
2006－UBI－10 (50)
2006／2／17

導電性布を用いたウェアラブルデバイスの機能再構成に関する検討

荒井 悟[a)] 戸田 真志[a)] 秋田 純一[b)] 岩田 州夫[a)]

a) 公立はこだて未来大学 b) 金沢大学

概要： ウェアラブルコンピュータ向けネットワークシステムであるTextileNetは，電源供給と相互通信の実装に伴うデバイスの配置自由度の損失を解消するために提案された．本稿では，TextileNetの機能再構成についてファームウェア更新に関わる煩雑な手順を問題と捉え，これを解決するためのアーキテクチャの提案，およびプロトタイプの実装と評価を述べる．このアーキテクチャは，ウェアラブルデバイスがTextileNetのデータ通信路を介してファームウェアを受け取り，自身のプログラムメモリを書き換えることを可能にするものである．これにより，ウェアラブルデバイスの配置自由度を損なわずに，TextileNetに機能再構成のためのシステムを組み込むことを可能にする．

A Discussion About Functional Reconstruction For Wearable Devices Using Conductive Fabric

Satoru ARAI[a)] Masashi TODA[b)] Junichi AKITA[a)] Kunio IWATA[a)]

a) Future University-Hakodate b) Kanazawa University

Abstract: TextileNet was proposed to resolve the problem about flexibility loss of spatial arrangement that is caused by laying power line and communication line. But TextileNet has a problem, which is cumbersome firmware updating. In this paper, we propose an architecture that resolves the problem. And we implement and evaluate prototype of the system using the architecture. The architecture allows wearable devices to download firmware from network of TextileNet, and the devices can update their program memory by themselves. Implementation of the architecture actualizes functional replacement for devices. And the architecture does not impair flexibility of arrangement in space.

1 はじめに

コンピュータを衣服のように身につけて利用するウェアラブルコンピューティングにおいては，電源供給路とデータ通信路を確保する方法が根本的な問題となる．すなわち，これらを架設する上で伴うデバイスの設置自由度の損失が，ほぼ全ての応用面において重要な問題として指摘されている．この問題を解決することが可能なネットワークシステムの一例として，戸田らは導電性衣服とバッジ型デバイスによって実装されたネットワークシステムTextileNetを提案した[1]．TextileNetは，衣服の表裏それぞれに導電性布を用いることで電極とし，このふたつの電極に接続されるウェアラブルデバイスへ電源供給とデータ通信を可能にするものである．しかしTextileNetの現在のシステムで問題と考えられるのが，煩雑

なファームウェア更新の手順である．本稿では上記の問題を解決するために，ウェアラブルデバイスのファームウェアを更新するアーキテクチャの基礎的検討について述べる．

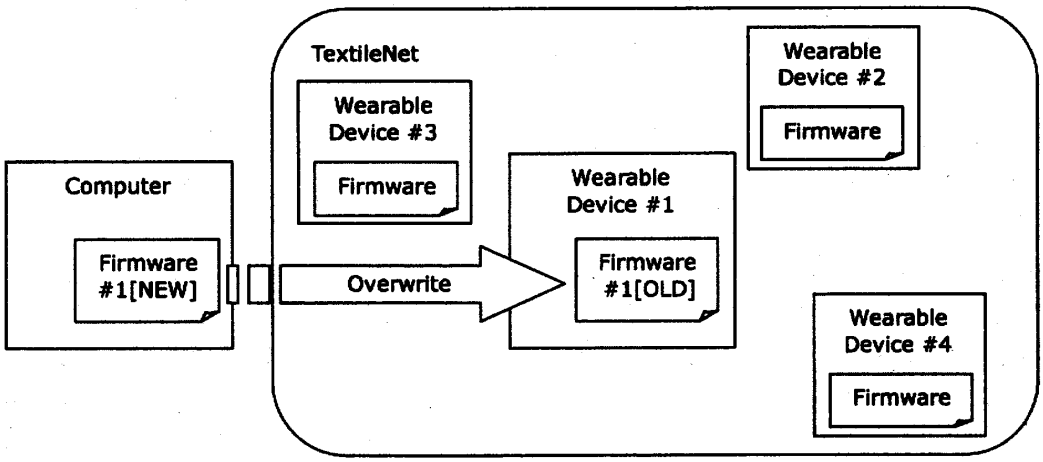

図 1 システム概略図
Fig.1 Schematic Of The Architecture

2 設計指針

本稿では，ふたつの電極でネットワークシステムに接続されるウェアラブルデバイスのファームウェアを，ネットワークシステムを介して更新可能にするアーキテクチャを提案する．すなわち，ウェアラブルデバイスがネットワークのデータ通信路を介して，外部に存在するコンピュータからデータを読み込み，ファームウェアを更新するものである(Fig. 1)．

デバイスの設置箇所や態様が自由にアレンジでき，個人が好みのシステムを構成することを可能にする TextileNet のシステムでは，その機能再構成に伴うファームウェアの更新は必要不可欠である．提案するアーキテクチャの導入により，この更新の手順がより容易に実行可能なものとなる．

また，ウェアラブルデバイスのファームウェアに不具合が見つかった場合にネットワークを経由して修正できるほか，後から機能の拡張ができることで，システムの機能維持などにも応用の可能性が期待できる．

設計にあたって，実装に伴う回路変更を最小限にとどめることを指針とした．回路規模の肥大を抑えることにより，デバイスの配置自由度を損なわずに実装の完了を目指すものである．

TextileNet におけるウェアラブルデバイスは 1 チップマイクロコンピュータである PIC マイコン[2]を使用して制御されており，ファームウェアは PIC マイコンのプログラムメモリに格納される．フラッシュメモリで構成される PIC マイコンのプログラムメモリは，通常 PIC マイコンプログラマと呼ばれる専用ハードウェアを使用して書き換える必要がある．しかしこの方式では，その都度 PIC マイコンプログラマのソケットに PIC マイコンを装着しなければならない．そのため，ウェアラブルデバイスに PIC マイコンを実装した後の更新の手順が煩雑になる．

この煩雑さを解消するための対応策のひとつとして，各々のウェアラブルデバイスに対し PIC マイコンプログラマと等価の回路を実装する方法が考えられる．しかし，この方法では前述の回路規模の肥大という問題を免れることができない．ウェアラブルデバイスの条件特性[3]のひとつとして挙げられる装着性を犠牲にすることは，アーキテクチャが解消する問題以上に大きな問題である．

そこで，PIC マイコンの持つ命令として用意されるプログラムメモリの書き換え機能を利用し，ネットワークから読み込んだ更新データでファームウェアを書き換えるプログラムローダを実装，これをウェアラブルデバイスに組み込むことで対応する．

3 実装方式

プログラムローダの実装は，Microchip Technology Inc.によるアプリケーションノート AN851[4, 5]で公開される仕様に基づくものである．この資料は PIC マイコンのブートローダ構築に関するものであるが，プログラムメモリがフラッシュメモリで構成される PIC マイコンを用いて，ソフトウェアによるプログラムメモリの自己書き換え機能を実装している．

この実装方式ではプログラムメモリ上にファームウェアとは別に保護された領域を設定し，それをプログラムローダ用の領域として割り当てる．そして，残されたファームウェア用の領域をプログラムローダが必要に応じて書き換える．

このプログラムローダをウェアラブルデバイスに適用するにあたり，ネットワークシステム外部のコンピュータから更新データを読み込むための機構が必要となるため，関連する仕様を追加の上で実装する．

3.1 ヘキサファイルの解析

ファームウェアの更新データは，インテルヘキサフォーマット[6]で記述されたファイルとして生成される．これはプログラムメモリ上のアドレスと格納されるバイナリ値，およびいくつかの関連情報を ASCII コードで表現したテキストファイルである．PIC マイコンのプログラムメモリ書き換え機能を用いる上で必要なのは，格納先のアドレス情報とバイナリ値を整数の形で取り出したものである．更新データを解析し，これらのデータを得るプロセスが必要となる．

ヘキサファイルではテキストの 1 行がひとつの情報単位でなる．よって，ヘキサファイルから 1 行ごとに読み出しと解析を逐次処理し，取り出されたデータを更新対象のウェアラブルデバイスへと送信する．ウェアラブルデバイスに組み込まれたプログラムローダは，そのデータを受信してプログラムメモリの書き換え命令を実行する．

3.2 プログラムメモリの書き換え

ウェアラブルデバイスに組み込まれるプログラムローダのデータは，プログラムメモリ上でファームウェアとは別の保護された領域に配置される(Fig. 2).

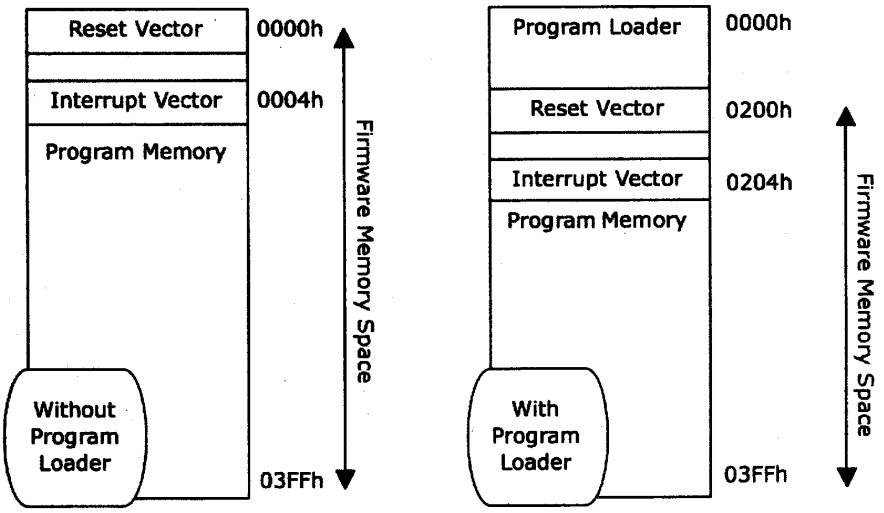

図2 プログラムメモリマップ

具体的には，0000h から 3FFF まであるアドレス空間で 0000h から 01FFh までの領域がプログラムローダ用の領域として設定される．ファームウェアのコードは 0200h 番地以降の領域に配置され，リセットベクタおよび割り込みベクタもこの領域に再配置される．

ウェアラブルデバイスは電源投入後，ファームウェア領域のコードを実行する．バックグラウンドではデータ通信路からの受信データを常時監視しており，ファームウェア更新の要求があると実行をプログラムローダ領域のコードに切り替える．プログラムローダは格納先のアドレス情報とバイナリ値のデータを受信し，データの整合性をチェックした後にプログラムメモリの書き換えを実行する．データの受信と書き換えの手順は，ファームウェアのデータが全て受信されるまで繰り返される(Fig. 4)．ファームウェアの更新が終了すると，プログラムローダはウェアラブルデバイスを再起動し，再び実行されるファームウェア領域のコードは更新後のものとなる．

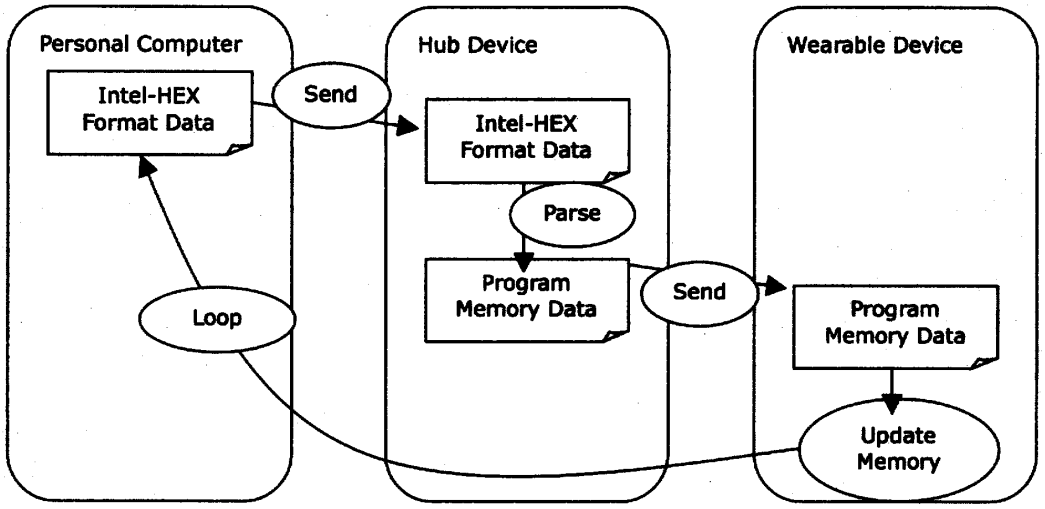

図3 プログラムロード流れ図
Fig.3 Program Loading Flowchart

3.3 データ通信

ヘキサファイルはネットワークシステム外部のコンピュータ上に存在するが，そのままではネットワークシステムの内部にデータを送信することはできない．表面電極の伝送路に全てのデバイスが接続される TextileNet のネットワークシステムでは，独自の通信規格で通信が行われるため，外部との接続用インターフェイスとしてネットワークハブとなるデバイスを設けて対応する．バス型の構成となるネットワークシステムの内部では，ハブデバイスは外部から受信したデータをネットワークシステムの汎用通信フォーマットでウェアラブルデバイスに一斉送信する．

4 プロトタイプ

プロトタイプとしてソルダーレスブレットボード上にシステムの実装を行い，その動作を確認した(Fig. 4)．導電性布を用いた実際の TextileNet と全く等価な環境を用いたプロトタイプ構築は，今後進めていく予定である．

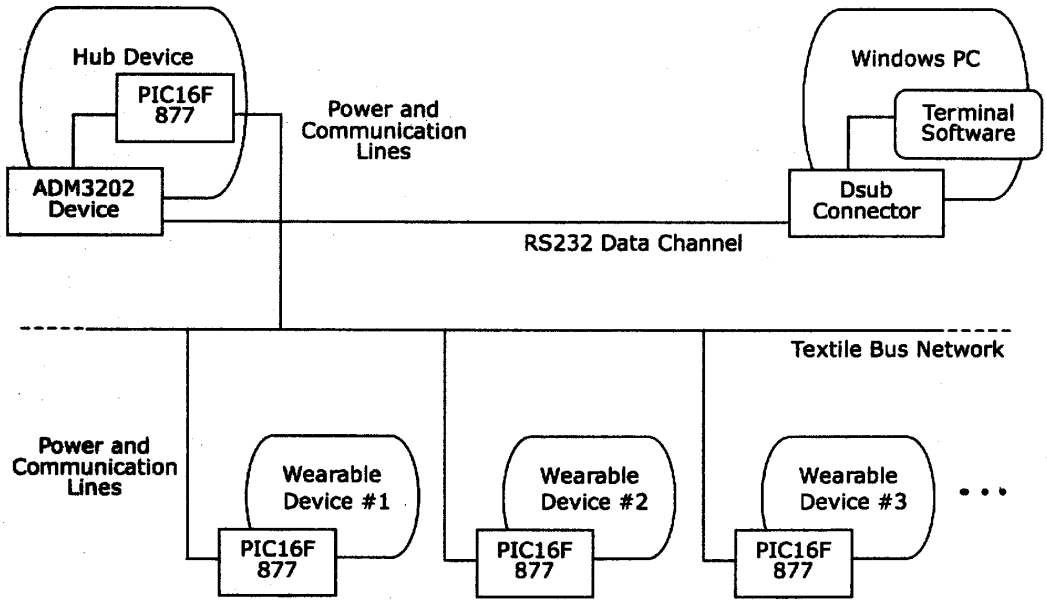

図4 プロトタイプ
Fig. 4 Prototype

4.1 ウェアラブルデバイス

ウェアラブルデバイスの制御用 PIC マイコンにはミッドレンジシリーズでフラッシュメモリ構成のプログラムメモリを採用する PIC16F877 を選択した.

PIC16F877 は数多くの入出力ポートを備えるが, このうちバス型構成のネットワークシステムに接続されるのは入力信号の変化に対して割り込み処理を実行可能な RB0 ピンである. ウェアラブルデバイスは常時 RB0 ピンからの入力データ, つまりハブデバイスから送信されるデータを監視し, 受信したデータにより対応する処理を実行する.

尚, ハブデバイスからウェアラブルデバイスへのデータ送信は一斉送信なので, ウェアラブルデバイスには各々に一意の識別番号を持たせておき, データの送信先と自身の識別番号を比較してデータの取捨選択を行う.

4.2 ハブデバイス

ウェアラブルデバイス同様に, ハブデバイスの制御用 PIC マイコンには PIC16F877 を選択した.

ネットワークシステムへの接続方式も, ウェアラブルデバイスと同様のものである. ウェアラブルデバイスとの相違は, 外部へ接続する RS232 シリアル通信インターフェイスの有無である. これは PIC16F877 の持つ USART 通信モジュールを利用し, 半二重通信. 受信用の RC7 ピンと送信用の RC6 ピンを 2 チャネル RS232 インターフェイスデバイスの ADM3202 に接続し, 汎用 9 ピン DSUB コネクタを取り出した.

ヘキサファイルの解析部分は, プロトタイプではこのハブデバイスに組み込まれている. ハブデバイスは 9 ピン DSUB コネクタから接続されたコンピュータのターミナルと通信し, ヘキサファイルの受信を行う.

4.3 ターミナル

ハブデバイスとシリアル通信するターミナルは, プロトタイプでは Windows PC のハイパーターミナルである. ターミナルからハブデバイスへファームウェア更新要求と更新対象

ウェアラブルデバイスの識別番号を送信すると，ハブデバイスはターミナルにヘキサファイルの送信を促す．併せて，この時点でハブデバイスは全てのウェアラブルデバイスに組み込まれたプログラムローダの起動を指示する．

ターミナルとハブデバイスの通信フロー制御は，ハブデバイスに組み込まれたヘキサファイル解析部分が担当する．プロトタイプではXON/XOFF文字を用いたソフトウェアフロー制御を用いており，ヘキサファイルの行単位での読み出しが行われる．

4.4 通信データ

ハブデバイスはヘキサファイルを解析後，格納先アドレスの情報とバイナリ値のデータをウェアラブルデバイスへ一斉送信する．この後，ファームウェア更新が終了するまでハブデバイスからウェアラブルデバイスへ送信されるデータは全て，38個の要素を持つbyte型の配列データである．具体的にはデータ先頭フラグと送信先ウェアラブルデバイス識別番号の2要素に，格納先アドレス情報とバイナリをbyte型の配列データに変換した35要素，そして誤り訂正符号の要素を加えた配列データである．

データの送信は1要素ずつ逐次処理で行われ，データ先頭フラグを受信する全てのウェアラブルデバイスは，次の要素である送信先ウェアラブルデバイス識別番号を受信する．この段階で各々のウェアラブルデバイスは自身の識別番号との比較を行い，一致した場合のみ以降のデータを受信し，データに誤りがなければプログラムメモリの書き換え操作を実行する．識別番号の比較が一致しなかったデバイスは，次のデータ先頭フラグが受信されるまで受信データをスキップする．これらの手順はファームウェアの更新が終了するまで繰り返し実行される．

ファームウェアの更新が終了すると，ハブデバイスは全てのウェアラブルデバイスに再起動を指示し，再びファームウェア更新要求があるまで待機する．

5 おわりに

本稿ではTextileNetの技術を基盤として，導電性布を用いたウェアラブルデバイスの機能再構成について検討し，現状の問題であるファームウェア更新の煩雑性を改善するアーキテクチャを提案した．プロトタイプの実装により，更新データをTextileNetのデータ通信路を介して読み込み，ウェアラブルデバイスのファームウェアを更新するシステムが実際に構築された．今後はこのプロトタイプを基盤として，実際にピンバッジ型ウェアラブルデバイスへの組み込みの検証を進める予定である．

参考文献

[1] 戸田真志，秋田純一，"空間自由度の高いネットワーク基盤に関する検討"，情報処理学会ヒューマンインターフェース研究会研究報告，2004-HI-107，pp.27-32，Feb. 2004.

[2] Microchip Technology Inc., "PICmicro Microcontrollers", Web, 2006, http://microchip.com/.

[3] Steve Mann, "Definition of wearable computing.", Web, 1980, http://wearcomp.org/wearcompdef.html.

[4] Ross M. Fosler, Rodger Richey, "AN851: A FLASH Bootloader for PIC16 and PIC18 Devices", Microchip Technology Inc., Aug. 2004.

[5] マイクロチップ・テクノロジー・ジャパン株式会社，"PIC マイコン デザインノート：PIC16F/PIC18F のブートローダ機能の紹介"，グローバル電子株式会社 Global News, No. 22, pp.27-28, Jul. 2004.

[6] Intel Corporation, "Intel Hexadecimal Object File Format Specification Revision A", Jan. 1988.