フィジカルコンピューティング・プラットフォームを使った実世界インタフェースの試作

桑田喜隆[1]
(株)NTT データ

Prototyping Real-world Interface with a Physical Computing Platform

Yoshitaka Kuwata
NTT DATA CORP.

概要：災害現場で利用される端末などの試作においては，ハードウエアおよびソフトウエアの試作と評価を繰り返しながら，スパイラルな設計方法が必要とされる．本稿では，新たなフィジカルコンピューティング・プラットフォームについて説明する．また，紹介したフィジカルコンピューティング・プラットフォームを活用し，災害活動時の安全確保のために作業員の転倒を検知するセンサを試作した事例紹介し，その有効性を論じる．

１．はじめに

災害現場で利用される端末など，通常のグラフィカルユーザインタフェースの適用が難しい分野では，キーボードやマウスを使うことが難しいため，通常のコンピューティングとは異なる要求が生じる．例えば，センサやアクチュエータなどの外部デバイスをPCに接続することが必要とされる．実世界のインタフェースを伴うため，多くのアプリケーションでは要求条件が予め明らかでない．このため，ハードウエアおよびソフトウエアの試作と評価を繰り返しながら，スパイラルな設計方法が必要とされる．

筆者は災害対応活動などで隊員の異常を検知するため，市販のウエアラブルコンピュータに転倒センサを接続し，無線 LAN を使って状況を転送するシステムを試作した[1]．隊員の異常を検知するための方法として，水銀スイッチによる簡易な方法を用いたが，本来であれば，身体の活動状態などを正確にモニタリングして的確な判断を行う必要がある．マイクロプロセッサにつないだセンサのファームウエアに転倒検知アルゴリズムを直接組み込んだため，開発の手間がかかり，簡単な実装のみ実施することとした経験がある．

他方，フィジカルコンピューティングは従来のキーボード／モニタ／マウスを使ったグラフィカルユーザインタフェースの枠組みを超えたユーザインタラクションの方法を研究する取り組みである[2]．インタラクションの追求のために素早くプロトタイプシステムを作成する必要があるため，複数のフィジカルコンピューティング・プラットフォーム(Physical Computing Platform, 以下 PCP) が提案されている．PCP を使うことで，プロトタイプ作成が容易になると考えられる．

本稿では，従来の PCP と互換性があり，より実現の容易な PCP を紹介する．また，PCP を活用し災害活動時の安全確保のために作業員の転倒を検知するためのセンサを作成する事例を報告する．

２．PCP を使ったプロトタイピング

プラットフォームとして，予めハードウエアおよびソフトウエアライブラリを用意しておくことで，フィジカルコンピューティングのための試作を容易にすることが可能となる．

図１に多くの PCP で採用されているアーキテクチャを示す．パソコンに外部からデータを入出力するためのデバイスとパソコン側の開発環境から構成される．本論文では，以下の物を含めて，PCP と呼ぶことにする．

[1] 桑田喜隆
(株)NTT データ
〒 134-0081 東京都江東区豊洲３－３－９
豊洲センタービルアネックス
kuwatay@nttdata.co.jp

・ フィジカルコンピューティング・デバイス
・ 上記のファームウエア
・ 関連するライブラリ
・ ソフトウエアの開発環境

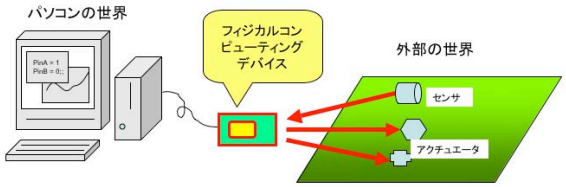

図 1 PCP のアーキテクチャ

PCP は大別して「ファームウエア変更型」と「外部制御型」に分けて考えることが出来る．両者の実現方法の比較を図２に示す．また，両者の実装の特徴を表１に示す．

前者はユーザがフィジカルコンピューティング・デバイスのファームウエアを開発し，デバイスにダウンドードして実行される．他方，外部制御型はデバイス側に汎用的なファームウエアを内蔵しており，ユーザはパソコン側からコマンドを通じて制御を行うプログラムのみを記述する．

前者の代表に，Arduino[3]があり，後者には Gainer[4]がある．Arduino はスタンドアロンでの利用が可能であるという特徴がある．一方，Gainer はハードウエアの詳細を知らなくてもコマンドを発行するだけで利用出来る点でメリットがある．Arduino は ATMEL 社の AVR チップを採用して，Gainer は CYPRESS 社の PSoC チップを採用しているという違いもある．どちらもホストとの通信には USB インタフェースが採用されている．また，オープンなライセンス形態をとっている点も表通しており，プロジェクト例も数多く公開されていることから，それらを参考にプロトタイプを素早く完成することが可能になると考えられる．

開発環境で特に注目すべき点は，初心者向けにプロトタイピングを容易にする Processing[8]が採用されている点である．Processing は Java を単純化した言語であり，”sketchbook”と呼ばれる統合開発環境を使うことでプロトタイプを容易に行うことが出来る仕組みを提供している．

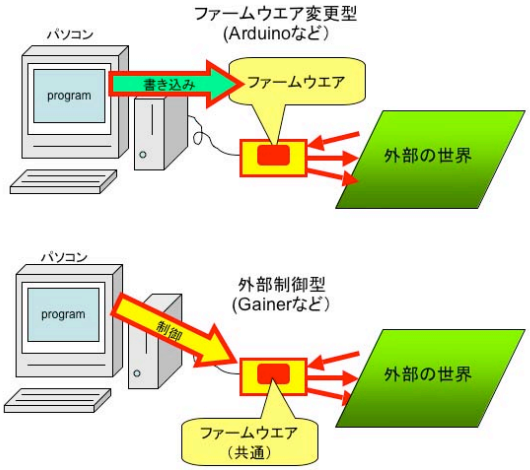

図２　PCP の実現方法の比較

表１ PCP の実装方法の特徴

	ファームウエア変更型	外部制御型
実行形態	スタンドアロン ＰＣとの連携	ＰＣとの連携
実装例	Arduino Wiring[5]	・Gainer ・USB Bit Whacker[6] ・Arduino with Firmata[7]
ファームウエア開発環境	Arduino (Processing)	(不要)
ホストＰＣ開発環境	Processing Adobe Flash, Max/MSPなど	Adobe Flash, Processing, Max/MSPなど

３．USB インタフェースをソフトウエアで実現した PCP 用デバイス

外部制御型の Gainer をベースにして，より手軽に利用可能な PCP 用のデバイスを設計した．以下に設計の方針を示す．

・ Gainer で採用されている PSoC に代わり，より高速な AVR を採用する
・ AVR の USB ソフトウエアスタックである AVR-USB[9]を採用し，USB インタフェースをソフトウエアで実現することで，部品数を減少する．
・ Gainer の既存の資産を活用出来るように，ホストとの通信プロトコルやピン配置等は出来るだけ Gainer と同じにする．
・ 制御対象の規模に応じてデバイスを選択出来るように，予め大中小３種類のデバイスを用意しておく．

表２にデバイスの概要を示す．Ginger は，Gainer と同じ 28 ピンのデバイスを使用しており，ほぼ Gainer と同様の機能を実装している．Pepper は 8 ピンのデバイスで小規模用途に特化している．Sugar は 40 ピンのデバイスを利用して 32 本の入出力を同時に利用可能である点が特徴となっている．

表２　試作デバイスの概要

名称	Ginger	Pepper	Sugar
入出力ピン数	16	4	32
アナログ入力数	6	3	8
アナログ出力数	6	2	6
デジタル入力数	16	0(*)	16(*)
デジタル出力数	16	2	16(*)
デバイス	ATMega88	ATTiny45 ATTiny85	ATMega164
コンフィグレーション数	7	3(*)	3(*)

(*) 暫定仕様

図３にデバイスの外観を示す．Ginger と Sugar は Gainer に倣ってブレッドボードに差し込んで利用し易いように，下部に入出力用のピンを配置している．Pepper は小型のセンサ等を接続し易いように，横向きのコネクタを実装する設計である．ブレッドボードで利用する場合には下向きのピンを配置することも可能である．

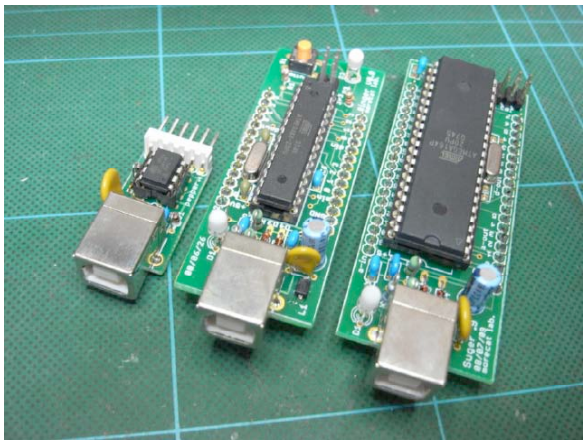

図３　USB をソフトウエアで実装した PCP 用デバイス
左から，Pepper, Ginger, Sugar

４．現場活動中の隊員の状態をモニタするためのセンサシステムの実現

ここでは，PCP による試作の場合と比較するため，筆者が過去に試作したセンサシステムの実装について述べる．

4.1 現場活動支援システムの概要

まず，筆者らの作成した隊員活動支援システムのプロトタイプの写真を図４に示す．

図４　隊員活動支援システムのプロトタイプ

本システムは，市販のウエアラブル PC と無線 LAN，GPS 等を組み合わせることで，隊員同士の位置情報に基づく情報共有を実現した．隊員の位置を把握しながら，指示を与えたり，隊員の集めた画像を地図上にマッピングし，状況を確認することが可能となる．

試作システムを利用して，災害対応現場のニーズのヒアリングを実施した．隊員の安全管理の面から，全隊員の状態をモニタリングし，危険な状態が検知された場合には，通知を行う仕組みを実現したいというニーズがあげられた．現場からは，隊員の呼吸状態や心拍情報などのバイタル情報を直接管理したいとの要望が上がったが，実装や運用上の課題が多いため，簡易な代替え手段として，隊員の転倒を検知して，通知する方法を検討することとした．

4.2 センサシステムの試作

隊員の状態把握のために試作した転倒センサの外観を図５に示す．センサとしてはガラスに封入された水銀スイッチ（画面左側）を採用した．基板が水平になった場合に，水銀スイッチの接点が離れ，導通のなくなったことをマイ

クロプロセッサで検出する．同じ状態が３０秒以上続く場合には，パラレルインタフェースを通じてホストに状態を通知する仕組みにした．

マイクロプロセッサには Microchip 社の PIC16F84 を採用して，ユニバーサル基板上に実装した．ファームウエアはアセンブラで記述してあり，他のライブラリなどは利用していない．LED の制御なども含めて，約５０行のプログラムである．

ハードウエアの設計，試作には約３日，ファームウエアの設計および試験は１日程度を要した．試作時には，PIC 向けの ISP(In Circuit Programmer)を利用できなかったため，ファームウエア開発時には，ソケットからプロセッサを外してプログラマに差し込んでファームウエアを書き込み，再度試作基板にプロセッサを戻して試験を行うという作業を繰り返す必要があった．またファームウエアの開発の開始時点においては，ハードウエアの動作が確認出来ていなかったため，ハードウエア試験用のプログラムを実行して動作確認を行う必要があった．

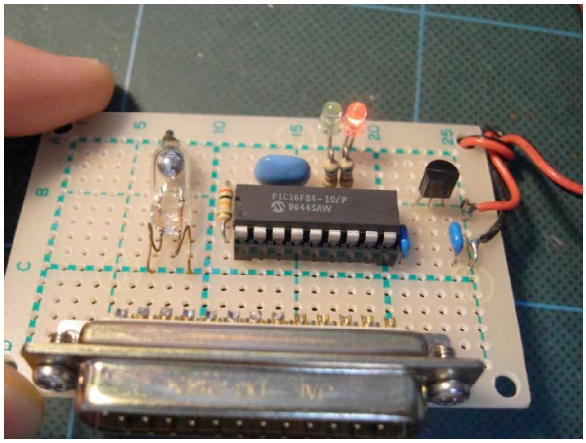

図５ 試作した転倒センサ

５．PCP を使った転倒センサの実装と評価

前章で述べた開発との比較のために，PCP を使って上記センサシステムに相当するシステムの試作を実施した．

センサを一つだけ使った小規模な実装であるため，試作には Pepper を利用した．入手の都合で，水銀センサの代わりに傾斜スイッチ[2]を利用した．また，３軸の加速度センサを使った評価も同時に行うことした．

5.1 PCP を使った実装（その１）

傾斜スイッチはポートに直接接続するだけで非常に簡単である．試作にブレッドモードを使えば，ジャンパ線で接続するだけで完了する．図６に実装状況を示す．

ホスト側は Gainer の Processing 向けライブラリに付属する例題から，アナログ３チャンネルの読み取りとグラフ表示プログラムを選んで，Pepper の仕様にあうように数行の手直しを行った．

転倒センサからの出力結果を図７に示す。

１チャンネルのみにセンサを接続しているため，グラフの３個目のみに値が表示されている．また，本来，傾斜スイッチはオンかオフの二値しか取らないため，デジタルで読み込むことが可能であるが，Pepper はアナログ読み込みしかサポートしていないため，アナログ読み込みを行った．

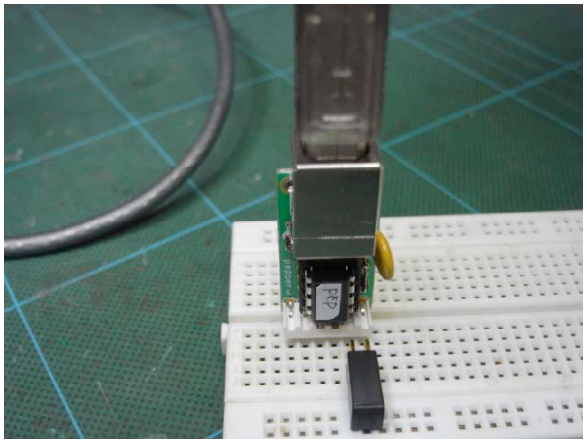

図６　PCP を使って試作した転倒センサ

[2] 傾斜スイッチは金メッキされたボールがセンサ内で移動することで傾斜の検出を行うものである．電気器具の転倒の検出等に使われている．

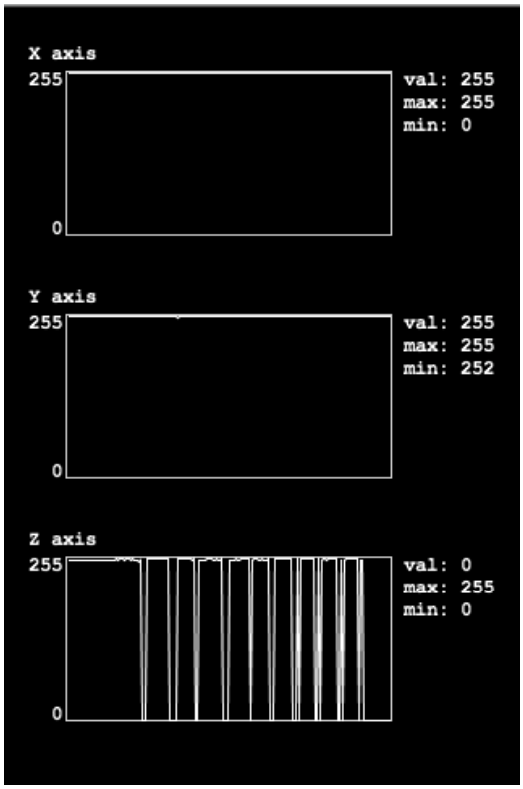

図７　転倒センサの出力結果

図７のグラフから，状態の変化時にスイッチにチャッタリングが生じていることが分かる．このため，実時間で状態を検出するためには，ソフトウエア的に処理を行うか，入力にヒステリシス特性を持たせる等の対策が必要となる．しかし，今回のアプリケーションの様に人間の姿勢を見るだけであれば，前回の試作と同様に一定時間転倒状態が続いていることを検出するだけで十分である．

なお，部品の調達を除いて，ここまでの試作にかかった時間は約１時間であった．

5.2 PCP を使った実装（その２）

次に，加速度センサを転倒の検知の目的で利用可能であるか検討するため，実際に Pepper に接続して動作させてみた．利用した加速度センサは，XYZ の３値が異なるピンからアナログ値として出力される．また，電源も 5V であるため，USB 経由でパソコンから供給される電源をそのまま利用することが可能であることが分かったため，従って，Pepper との接続は該当する端子間を繋ぐだけで，非常に簡単である．

図８に接続の様子を示す．比較評価のために，ブレッドボードに代わってユニバーサル基板に半田付けで実装した．

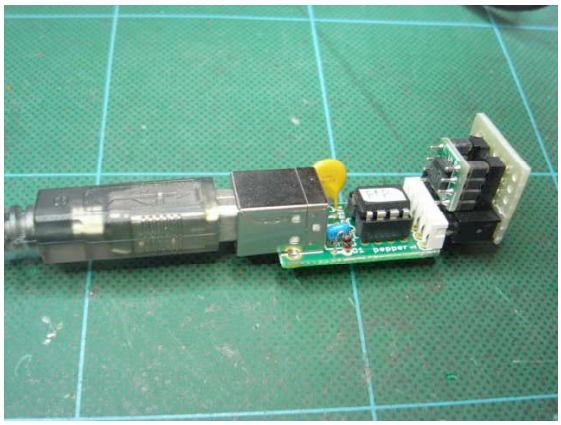

図８　３軸加速度センサの接続

図９に３軸加速度センサの出力結果の例を示す．5.1 章で述べた実装（その１）と同じプログラムを使うことが出来たので，プログラミング作業は不要であった．

転倒センサに比較して，アナログ値が取得可能であるため，３軸の組み合わせから姿勢の情報を推定することが可能である．また，反応性も良いことから，リアルタイムのアプリケーションにも向くと思われるが、取得データを基に姿勢の計算が必要となることが分かった．

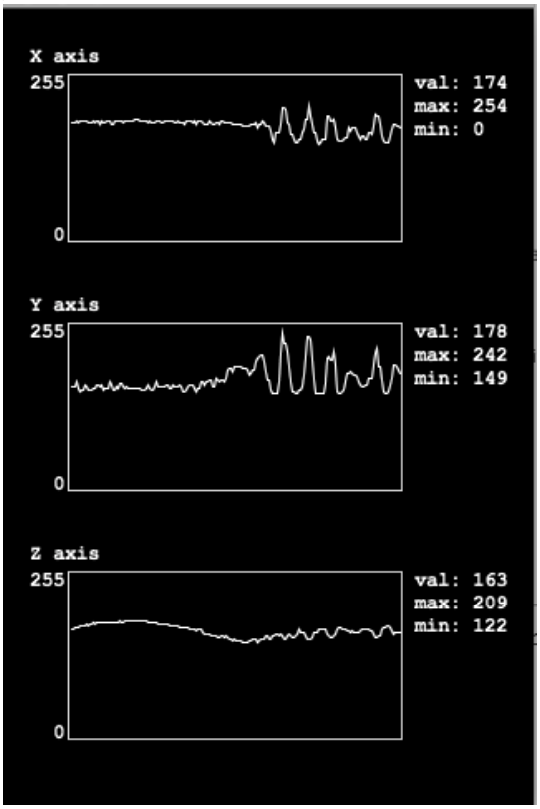

図９　３軸加速度センサの出力結果

5.3 考察

（1）開発効率の比較

PCP 無しにアセンブラで開発した場合には，4 日間（約 20 時間），今回 PCP を使って 2 種類のセンサを試すために約 2 時間を要した．開発の時期や条件が異なるため，単純な比較は困難であるが，PCP を使った場合と使わなかった場合では，開発にかかった期間に 10 倍の差があったことになる．以下の点で工数が削減されたと考えられる．

- ハードウエア開発
- ファームウエアの開発
- ホスト側のアプリケーション開発

いずれの要素も，新規に開発を行うのではなく，既存の資産が生かせたことの効果が最も大きい．次に，Processing による開発サポートによって工数が削減出来たと考えられる．

（2）柔軟性

PCP を利用した場合には，試行錯誤を気軽に行えるメリットがあった．当初は加速度センサの評価まで実施する予定ではなかったが，試しに繋いでみるということが簡単に行えるため，手軽に実施することができた．PCP を活用することで，短時間に複数の異なるアプローチを比較検討することか可能になった．

（3）スタンドアロン型の必要性

評価に使った PCP は Gainer のフレームワークをそこまま利用しているため，基本的にはホストからのコマンドで動作する[3]．

アプリケーションによってはデバイス側で処理を実行し，ホストの負荷を減らしたい場合もある．例えば，今回の加速度センサの実装では，デバイス側で転倒の判断を行い，転倒したかどうかの情報のみをホストに送ることでホスト側の処理を軽減することが可能である．

また，ホストなしで単独実行可能なデバイスとして実装した方が良い物もある．このため，デバイス側で動くプログラムをユーザで登録可能なようにするなどの，フレームワークの拡張が望まれる．

[3] コマンドーレスポンス型のプロトコルであるが，連続入力モードは一度ホストからコマンドを送るとデバイス側から定期的に値を返すことが可能である．

６．まとめと今後の課題

本稿では，PCP を活用した実世界インタフェースの試作について述べた．また，実例として，転倒センサの実装について取り上げた．試作を通じての評価では，PCP を使うことで大幅に試作に要する時間が削減されることが分かった．特に，設計の途中で試行錯誤を行うような場合には有効である．

本稿で説明した PCP は公開されているため，多くの分野での活用が期待される．

参考文献

[1] 桑田喜隆，神成淳司，吉田茂樹，大谷尚通，井上潮，現場活動支援のための地理情報に基づく実時間情報共有システム，情報処理学会グループウエアとネットワークサービス研究会，41-4，2001 年 10 月 18 日

[2] Scarlatos, L. and Scarlatos, T. 2005. Physical computing and multimodal input in human-computer interfaces. J. Comput. Small Coll. 20, 5 (May. 2005), Pp.8-14.

[3] Arduino: http://arduino.cc/

[4] Wiring: http://wiring.org.co/

[5] Kobayashi, S., Endo, T., Harada, K., and Oishi, S. 2006. GAINER: a reconfigurable I/O module and software libraries for education. Proceedings of 2006 Conference on New interfaces For Musical Expression (Paris, France, June 04 - 08, 2006). Pp.346-351.

[6] UBW (USB Bit Whacker) Project, http://greta.dhs.org/UBW/

[7] Firmata, generic protocol to communicate between Arduino and host PC, http://www.arduino.cc/playground/Interfacing/Firmata

[8] Shiffman, D., Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction, Morgan Kaufmann Series in Computer Graphics

[9] Objective Development Software, AVR-USB, http://www.obdev.at/products/avrusb/index.html

本稿に記載されている会社名，商品名，またはサービス名は各社の登録商標または商標です．