

バーサタイルコントローラーの開発へ向けた構成要素の検証

4年C組 羽鹿 諒

指導教員 米田 隆恒

1. 概要

私は、以前「ウェアラブル・コントローラー」というものを開発した。現在行っている研究はこれをバージョンアップし、より実用性のあるものを開発することを目標としている。現在、この研究は「バーサタイルコントローラーの開発」として研究中である。よってここでは、現在開発中のハードウェアを構成する要素の検証結果を報告する。

キーワード コントローラー、SH2 マイコン、音声認識、モーショントラッキング、IMU、無線通信

2. 研究の背景と目的

以前私は「ウェアラブル・コントローラー」(図1)に関する研究を行っていた。



図1 ウェアラブル・コントローラー

これは、私たちが機械を操作する際、より直感的に、より簡単に操作できるように

と考えて開発したコントローラーである。

用いる際は腕に本体を、指にセンサ部分を装着し、ボタンを押しながら空中に図形を描く。すると、シリアルケーブルで接続した先のPCの画面に認識できた図形の名前が表示されるというものである。動作中の風景を図2に示す。



図2 動作中風景

原理を簡単に解説する。まず、指に装着しているセンサは加速度センサである。空中に図形を描く際、図形の頂点にさしかかると加速度センサから出力される信号には

大きな変動が生じる。この信号の変動をコントローラーは図形の頂点として読み替え、その頂点の数をカウントする。私は、図形の形を決める要素がその図形の頂点の数にあるということに気づいた。そこで、カウントした頂点の数をもとにユーザーが描いた図形を識別した。

しかし、ウェアラブル・コントローラーはあくまで図形の名前を取得するにとどまっており、実用的なものではなかった。

そこで、今回完全に別のハードウェアとして1からコントローラーを開発し直すことにした。より実用性のある新しいハードウェア、ユーザーインターフェイスを開発するためである。なお、この研究は現在も研究中であり、ここでは新しいコントローラーであるバーサタイルコントローラーを開発することを目標に、それに向けて必要な構成要素を検証した結果を報告する。

3. 研究内容

(1) 研究事項

【研究1】 信号処理の心臓部であるマイクロコントローラー(以下 MCU という)について

【研究2】 コントローラーの動きを計測するセンサについて

【研究3】 コントローラーが扱うアプリケーションについて

【研究4】 コントローラーの無線通信について

(2) 実際に行った研究

【研究1】

これまでの研究において、また、以前開発したウェアラブル・コントローラーにお

いても H8 という MCU を使ってきた。この MCU の優れている点として、入手性が高いことや国産の MCU としては安価に購入できること、使いやすいことなどが挙げられる。

しかし、この MCU は処理速度(計算速度)が遅い。一般に、MCU や CPU などの演算スピードを計測するためにベンチマークテストと呼ばれるものがある。これは、あらゆるプログラムのエッセンスを抽出し、作られている。いわばテスト用ソースコードのようなものである。これを用いて実際に MCU の処理速度を計算した結果がメーカーサイトに掲載されており、H8 は約 2.3MIPS となっている。MIPS とは、Million Instructions Per Second の略で、1秒間に何百万個の命令が実行できるかを表したものである。MIPS は、コンピューターの処理速度に関する情報でよく使われる。H8 のこの数値は、同レベルの MCU よりは高い方であったが、バーサタイルコントローラーを実現するには、さらに高い計算能力が必要であると考えている。また、H8 はコマンド環境(CUI)でしか開発ができなかった。図3に CUI の例を示す。

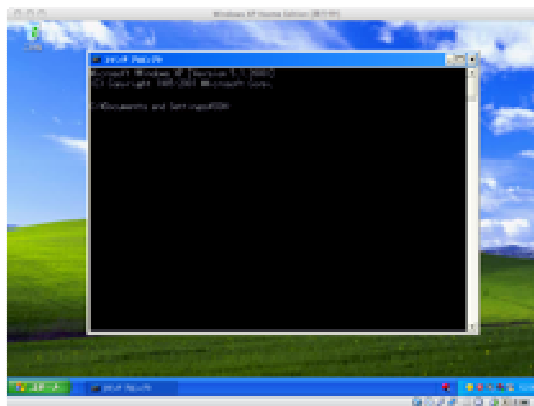


図3 コマンド環境

コマンド、つまりコンピューター用の単語のようなものでしか MCU の開発ができないため、MCU の細部の調整や処理速度の向上などを行うことが難しい。これは、MCU 上で処理を行うアプリケーションにとっては致命的である。そこで、新しい MCU として SH-2 マイコンに目を付けた。また、その中でも H8 と MCU 本体が同じサイズである SH7125F というものをピックアップした。このマイコンの特徴をあげると、まず処理速度が H8 と比較して格段に速いということがある。図 4 に表を示す。

H8 tiny	SH2
2.3MIPS	65MIPS

図 4 演算性能比較

さらに、開発環境が充実していることが挙げられる。まず、E10A-USB というデバッガを用いて開発することができる点である。E10A-USB の画像を図 5 に示す。



図 5 E10A-USB

これは、MCU 内部にアクセスして処理速度を優先するかプログラムの大きさを優先するか、など様々な設定を行いながら MCU を効率よく動かすために必要である。

また、HEW(図 6 : High-performance Embedded Workshop)とよばれる統合開発環境の存在も大きい。

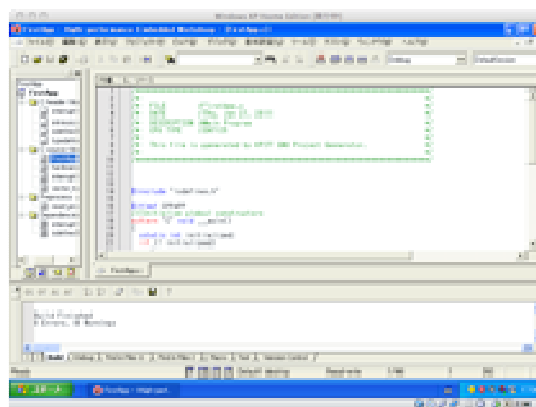


図 6 HEW 画面

これと E10A デバッガを組み合わせることで、プログラムの 1 行単位で MCU を動作させたり、MCU をチューニングしたりすることが可能となる。現在は、市販されている SH7125 ボードを用いてオリジナルのテストボード(図 7)を作製し、実験開発を行っている。なお、使用している開発用のプログラムコンパイラは、フリーで公開されている KPIT GNU Tools という、GCC(以下に詳細を記す)ベースのものを用いている。



図7 オリジナルテストボード

さらに、私は今回 Unix 環境でのプログラム開発を行えるよう、独自で C コンパイラおよびデバッガをビルドし、パッケージ化することを行った。上記のような、Windows で用いることができるパッケージはあったものの、Unix 環境で用いることができるものはなかったからである。今回は、インターネット上にフリーで公開されている GCC(GNU Compiler Collection) や GDB(GNU Debugger)、さらに組み込み MCU 向けの開発環境を提供するライブラリである Newlib, binutils といったソースコードパッケージを組み合わせ、MacOSX(v10.5、Leopard)上でビルドした。図8に MacOSX 上でコンパイラを動作させている様子を示す。



図8 コンパイラを動作させている様子

[研究2]

以前開発していたウェアラブル・コントローラーでは、加速度センサ(図9)のみを用いてプロトタイプを作製していた。

しかし、これには問題があった。それは、正確な運動が取得できないということである。ウェアラブル・コントローラーに用いていたのは、2軸(x, y 軸方向)の加速度センサであったため、上下の手のぶれや、微振動を検知することができなかった。また、加速度センサの信号の大きな変化しか捉えていなかったことも問題点である。つまり、頂点の数をカウントしていただけなので、三角形を描かなくても、うまく3回大きな振動を加えてやるだけで三角形と認識していたのである。この問題は、その他の図形でも同様に起こり得る事象である。



図9 加速度センサ

そこで、私はジャイロセンサに着目した。これは、物体の傾きを計測するセンサで、一般的には角速度を計測するものである。このジャイロセンサと加速度センサを組み合わせることで、正確な物体の動きをハードウェアでトラッキングすることができる。さらに、本来は物体の傾きを知るためのセンサなので、そのまま傾きと加速度を計測

したり、どのぐらいの傾きかを角度として計測したりすることも可能である。図 10 に一般的なジャイロセンサ素子の写真を示す。



図 10 ジャイロセンサ素子

今回、試験的にセンサモジュールを作製した。これは一般に、IMU(慣性センサ)という名前で知られている。使用したセンサは以下の通りである。また、作製したセンサモジュールの写真を図 11 に、センサ部を拡大した写真を図 12 に示す。

- ・ 加速度センサ:ADXL335
- ・ ジャイロセンサ:IDG-650(x, y 軸)
- ・ ジャイロセンサ:ISZ-650(z 軸)

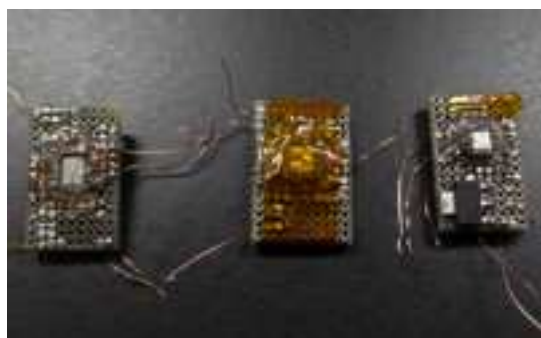


図 11 作成した IMU モジュール



図 12 センサ部拡大写真

現在は MCU に接続してセンサからの出力を読み込ませ、数値化したデータを無線で PC へと送信する実験を行っている。

[研究 3]

私はウェアラブル・コントローラーを開発する以前に、自作ハードウェアのみで人の発する声、母音を認識するという研究を行っていた。この研究は難易度が高すぎたために一度断念していた。しかし、今回の研究を行うにあたり、音声認識について調査を行った。対象としたのは、音声認識の原理と音声認識プログラムの開発に関するものである。

まず、音声認識プログラムについては、Julius(ジュリアス)というものがある。Julius は、京都大学の河原研究室が中心となって研究、開発を行っているオープンソースの汎用大語彙連続音声認識エンジンである。ここではシステムの構造が非常に難しいので詳細な説明は割愛するが、音声認識の世界ではよく用いられる確率モデルの 1 つである、隠れマルコフモデル(Hidden Malcov Model、HMM とも言う)を用いてシステムが構築されている。ソースコードから公開されており、ユーザーのシステム上でビルドし、動作させることが可能であ

る。低スペックマシンでインターネット接続環境がなくても使えることや、完全にシステムがフリーであることが利点として挙げられる。音声認識の原理を簡単に解説する。一般的な音声認識システムにおいてよく用いられるアルゴリズムは、隠れマルコフモデルに基づく「ビタビアルゴリズム」というものである。これは、観測された事象系列を結果として生じる隠された状態の並びの中にある最も確からしい並びを探すというものである。

また、音声認識は原理が複雑だけでなく、ソフトウェアの容量や必要とするシステムも MCU レベルのものでは到底できないことがわかった。Julius の紹介において低スペックのマシンでも動作すると記したが、これは MCU の性能に比べれば遥かに高いものだ。例えば、Julius を例に挙げると、音声認識を行うのに最低限必要なシステムパッケージで約 30MB ある。これは一般的な「マイコン」と呼ばれる IC に搭載できるシステムサイズより遥かに大きい。そこで、図 13 のような方法を考えた。また、以下のシステム案では無線通信の方法として、Bluetooth を例として解説する。

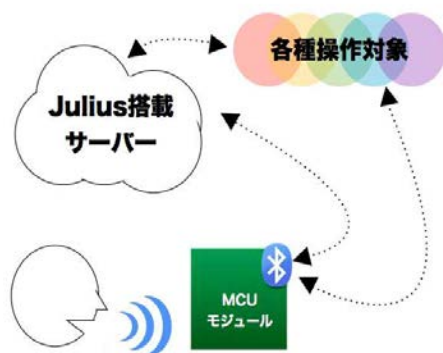


図 13 ハードウェア音声認識システム案

動作手順は、まずユーザーが MCU モジュール上のマイクに話す。この MCU モジュールには Bluetooth 通信機能がついており、外部の機器との無線通信が可能である。この Bluetooth 機能を通して Julius が動作しているサーバーにアクセスする。このとき、MCU モジュールで取得した音声データをサーバーへと送り、サーバー上で音声認識を行う。その後、サーバーから直接、もしくは MCU モジュールに一旦結果を返してから操作の対象となる機器に Bluetooth 経由で接続し、操作するというものである。

しかし、MCU の音声入力(厳密には A/D 変換機能)は音声認識に適するレベルに達していなかったため、これは使えなかった。一般的に、音声認識を行う際には 16 ビット、16KHz という分解能で音声データを読み込むことが必要とされる。

次に、ウェアラブル・コントローラーでも行ってきた、人の動きを解析するモーションセンシングについて説明する。今回の研究を行うにあたり、以前より精度を向上させることを目標にした。ソフトウェアについては現在研究を進めているので、ハードウェアの面については[研究 2]を参照されたい。ここでは、モーションセンシングを応用したアプリケーションに関して解説する。

まず、遠隔操作できる黒板を考えている。システムを図 14 に図示する。



図 14 遠隔操作黒板

ここでは、教室を例として解説する。このシステムを用いるには、教室にスクリーンとプロジェクター、Bluetooth 通信機能が搭載された PC が必要である。動作手順は、自作ハードウェアを用いて空中に図形や文字を描くと、その動きをハードウェアで分析し、Bluetooth を介して PC に送信し、プロジェクターから投影されるというものである。非常にシンプルであるが、ハードウェアのみで実際に手元の動きを解析し、再現できるかどうかは難点である。ただ、これが実現すれば教室中のどこからでも黒板にメモを残したり、板書をしたりといったことが可能である。次に、ホームコントローラーである。システムを図 15 に図示する。

システムの流れは、ここでは普通の家庭を想定する。このシステムも基本的には遠隔操作黒板と同様である。PC に Bluetooth を経由してハードウェアからの信号を送る。ただ、図 15 にもあるように、ホームコントロールという名前の通り、ドアや窓などの建具や、家電製品などを動かせる必要がある。そこで、別にコントロー

ル回路を作製し、高出力のモーターやソレノイドなどのアクチュエータで制御することになる。

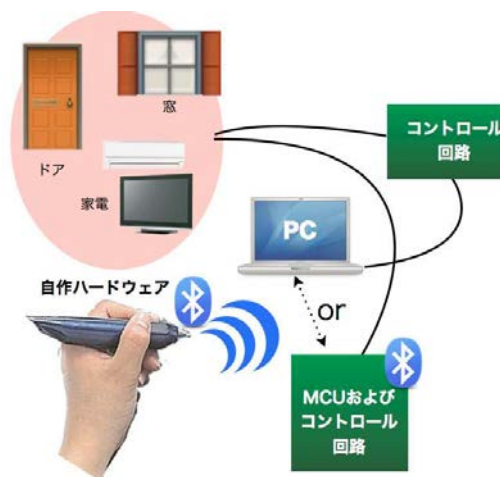


図 15 ホームコントロールシステム

これが実現すると、家中の家電製品や家の建具を簡単に動かすことができるのではないかと考えている。

さらに、加速度センサやジャイロセンサは本来運動量を計測するためのセンサである。よって、物理の授業などで使える教材に応用できるのではないかと考えた。それをまとめたものが以下の案である。システムを図 16 に図示する。



図 16 モーションセンシングシステム

このシステムの構造はここまでに解説した 2 つのシステムとほぼ同じである。しか

し、ここでは、自作ハードウェアをコンパクトに作る必要がある。また、用途によっては PC 側でソフトウェアを用意し、物理エンジンなどを組み込んでアニメーションを表示することも必要であると考えられる。

使用用途としては、ボールや振り子などにハードウェアを取り付け運動の様子を観察することなどがある。これが実現すると、目視やストップウォッチを見ながら行うような精度の悪い実験ではなく、精度の良い正確な実験を行うことができると考えられる。

[研究 4]

無線通信については、世界中で様々な種類のものが用いられている。また、各国で細かく規格が決められている。特に日本では、電波法の制限によって私たちが自作ハードウェアに用いることができる無線通信モジュールはかなり種類が限られてくる。

今回、私は 3 種類の通信規格について調べ、実際にモジュールを実験できたものは実際に通信テストを行い、それぞれのメリット、デメリットを検証した。

まずは、無線 LAN である。近年、スマートフォンやネットブックの流行によって一般のユーザーにも Wi-Fi 通信という名前で浸透してきている。私たちが使うことができる無線 LAN モジュールとして、CF(Compact Flash)という規格のコネクタがついているモジュールが存在する。図 17 にモジュールの例を示す。



図 17 無線 LAN モジュール

今回、実験することはできなかったが、一般に無線 LAN は 10~30m 程度の通信をすることが可能であるとされている。汎用性が高く、通信速度も速い上、特にこのモジュールは MCU に直結することができるため、非常に有用である。

しかし、[研究 3]に示したように、小型のモジュールにおさめなければならない場合はこのようなサイズのアダプタは使えない。また、消費電力が他の無線通信方法よりも大きいため、小型のハードウェアには実装することが難しい。

次に、Bluetooth モジュールである。Bluetooth モジュールの例を図 18 に示す。



図 18 Bluetooth モジュール

今回、このモジュールについても実験することができなかったが、一般的に無線 LAN と同様に、10~30m 程度の通信が可能とされている。また、多チャンネル同時通信が可能なことでも知られている。つま

り、1つのポイントに対して複数のデバイスが通信する、いわゆるブロードキャスト通信が可能なのである。また、近年はPCや携帯電話、家電製品にも多く取り入れられるようになったため、開発さえできれば多数の機器と通信をすることができる。さらに、モジュール本体が小型なので、自作ハードウェアなどに簡単に組み込んで使うことができる。

しかし、このようなモジュールは入手性があまりよくない。また、機械を少ししか作らない場合はこのモジュールでコストが大幅にかかってしまう場合がある。また、場合によってはチップのファームウェアをオリジナルにカスタマイズする必要があるため、高度な技術が必要なことがある。

最後に、ZigBee モジュールである。図 19 にモジュールの例を示す。



図 19 ZigBee モジュール
(写真は XBee モジュール)

今回、実験できたのがこのモジュールである。これはハードウェアを自作する上で、無線化が必要な場合によく使われるモジュールである。ZigBee 自体は、家電製品向けの短距離無線として知られている。通信距離は Bluetooth 同様、10~30m 程度である。消費電力はきわめて少なく、乾電池

1本で数ヶ月~数年動作できるほどである。

ここでは、図 20 のような実験を行った。



図 20 XBee 実験概要図

簡単に説明するとまず、2つの XBee モジュールを用意し、片方には電池をつなぎ持ち運びができるようにする。もう一方は PC につなぎ、電波強度および電波の届く状況を観察した。なお、今回はどの程度届くかのみ調べた。場所は、本校の理科講義室、理科講義室前の廊下、理科講義室の前の廊下の曲がり角の地点である。

まず理科講義室は電波が届いた。これにより、普通の教室や家の部屋のなかでは電波が届くことになる。続いて、理科講義室前の廊下である。この地点は最初、電波強度が弱いため電波が届かないことが懸念されたが、結果的には廊下の端から端まで約 60m 電波が届いた。

しかし、唯一理科講義室前の廊下の曲がり角は電波が届かなかった。距離は 2 m ~ 3 m 程度しか離れていないのだが、校舎が鉄骨で組み立てられているからか、電波が届かなかった。

事前に調べた結果でも、手軽に使えて安価な反面、電波が届きにくく、長距離の通信や別室同士の通信などには向いていない

ことがわかっていた。だが、予想以上に電波が届かなかった。

3種類のモジュールを比較して考えたが、同室内なら通信ができることや、安価で小型であることから現在は XBee モジュールを用いて無線通信の実験を行っている。

4. まとめと今後の課題

今回は長期間にわたり、さまざまな研究要素の実験を行い、それぞれの要素を調べていった。今回の実験結果をもとに、本研究につなげていきたい。

5. 参考文献

- [1] 「ADXL335 データシート」、アナログデバイセズ社
- [2] 「SH7125F マイコンボード」、秋月電子通商
- [3] 「SuperH™ファミリ用 E10A-USB エミュレータ ユーザーズマニュアル 別冊 SH7125、SH7124 ご利用時の補足説明」、ルネサスエレクトロニクス社
- [4] 「IDG-650 Dual-Axis Gyro Product Specification」、インベンセンス社
- [5] 「ISZ-650 Single-Axis Z-Gyro Product Specification」、インベンセンス社
- [6] 「SH7125 グループ、SH7124 グループ ハードウェアマニュアル」、ルネサスエレクトロニクス社
- [7] 「SH7125 CPU ボード M3A-HS25 ユーザーズマニュアル」、ルネサスエレクトロニクス社
- [8] 「STK-7125 ハードウェアマニュアル」、アルファプロジェクト社
- [9] 「STK-7125EVB ハードウェアマニュアル」、アルファプロジェクト社

[10] Julius 公式サイト、
<http://julius.sourceforge.jp/>

6. 謝辞

今回の研究を行うにあたり、サイエンス研究会物理班顧問の米田先生には研究に関するアドバイスをいただきました。

音声認識や研究の進め方など、研究の行い方については(株)国際電気通信基礎技術研究所(ATR)の野間春生さん、多田昌裕さん、吉田俊介さん、水戸和さん、荻田紀博さん、平田康夫さん、足立隆弘さんをはじめ、多数の研究員の方にご教示いただきました。

また、SH2 マイコンおよび E10A デバッガの取り扱いについては、(株)たけびしでシステム開発を行われている方々にご教示いただきました。

この場を借りて御礼申し上げます。