特別研究論文

# 手袋型デバイスを用いた
# 自然的インターフェースの研究

平成 27年 2月 13日

神戸市立工業高等専門学校専攻科
電気電子工学専攻
高田崚介

# 英語要約

To operate a smartphone, we used some gestures with touch panel or using some voices with voice control. However, these systems have some problems. For example, while people is walking and using touch panel, they can’t be looking around. It’s dangerous. Then, voice control system has some weak points, too. people who is in a noisy place or library, can’t use voice control correctly. In order to solve this problems, we develop the new interface system, ”GROVE”. GROVE was used a data glove for operating a smartphone. ”GROVE” is coined word by ”grow up” and ”glove”. This system can detect various seosor datas, and send to a smartphone by bluetooth. Smartphone analyze these datas for operating some applications. we had already develop this system. However, GROVE has some weak points. Then, we developed new GROVE system which is using conductive fiber to get fingers bend. As a result, system had become more lighter than before one and reducing the cost.

# 研究発表一覧

1) 高田崚介, 若林茂 :「手袋型デバイスを用いた自然的インターフェース」，電気学会 関西支部 高専卒業研究発表会，2013.3.2

2) 高田崚介, 若林茂 :「手袋型デバイスを用いた自然的インターフェース」，産学官技術フォーラム’13，pp.81，2013.11.20

3) 高田崚介, 若林茂 :「手袋型デバイスを用いた自然的インターフェース」，教育システム情報学会 2013年度学生研究発表会，pp.27-28，2014.3.7

4) 高田崚介, 若林茂:「手袋型デバイスを用いた自然的インターフェースの研究」，ヒューマンインターフェース学会研究報告集，Vol.16，ヒューマンインターフェース学会，pp33-34，2014.3.26

# ー 目 次 ー

**第１章 概要** **iii**

**第２章 解説** **1**

２.１ 指文字 . . . . . 1

２.２ ハンドジェスチャーの検出手法 . . . . . 2

２.２.１ 画像処理を用いた手法 . . . . . 3

２.２.２ 3D深度センサを用いた手法 . . . . . 3

２.２.３ EMGを用いた手法 . . . . . 4

２.２.４ 光ファイバーを用いた手法 . . . . . 5

２.２.５ 機械的手法 . . . . . 5

２.２.６ ひずみゲージを用いた手法 . . . . . 6

２.２.７ 高密度触覚センサ . . . . . 7

２.２.８ 導電繊維を用いた手法 . . . . . 8

２.２.９ 検出方法の比較 . . . . . 9

**第３章 システムの構成** **10**

３.１ GROVEシステムのコンセプト . . . . . 10

３.２ ハードウェアの構成 . . . . . 11

３.２.１ 全体の構成 . . . . . 11

３.２.２ 2012年度版 . . . . . 12

３.２.３ 2013年度版 . . . . . 14

３.２.４ 2014年度版 . . . . . 16

３.２.５ 各年度の比較 . . . . . 17

３.３ ソフトウェアの構成 . . . . . 18

３.３.１ 全体の構成 . . . . . 18

３.３.２ Arduino側 . . . . . 19

３.３.３ スマートフォン側 . . . . . 20

**第４章 まとめ・結果** **22**

４.１ ジェスチャーの検出 . . . . . 22

４.１.１ GROVEシステムビューの解説 . . . . . 22
４.１.２ GROVEへのBluetooth接続 . . . . . 23
４.１.３ Wi-Fiカメラとの接続 . . . . . 24
４.１.４ ジェスチャーの検出 . . . . . 25
４.１.５ 測定精度 . . . . . 26
４.２ ロボットハンドの操作 . . . . . 27

**第５章　考察** **28**

謝辞 . . . . . 31
参考文献 . . . . . 32

**付 録A　付録** **33**

付録 . . . . . 33

# 第１章　概要

昨今, コンピュータやスマートフォン等の高性能な多機能電子機器はあらゆる場所で用いられるようになっている. コンピュータが一部の研究者のみが使用する機器であった時代においては, 膨大な物理的スイッチを用いてデータの入出力を行っていた. 現在, これらの電子機器を簡単に操作することができるのは, キーボードやマウス等の適切なUI（ユーザーインターフェース）設計があるからである. 本研究では, 特にスマートフォンのUIに注目した. スマートフォンで用いられているタッチパネルや音声操作等のUIは, 特にNUI(自然的インターフェース)と呼ばれ, 画面上で指を滑らせたり, 言葉を発するといったジェスチャーを用いて機器を操作することができるインターフェースである. これらの人間の直感に沿ったインターフェースは利便性が高いが, 同時に欠点も存在する. 例えばタッチパネルの場合, もし外出先で歩行している際にタッチパネルを操作していた場合, 手はふさがり, 小さな画面に注視しながら操作しなければならないため, 交通事故や犯罪に巻き込まれる可能性が高くなる. 実際に2013年5月には, スマートフォンを操作していた小学5年生の男児が誤って駅のホームから転落し, 怪我をするという事故が起こっている[1]. また, 音声操作は周囲が騒がしい場所ではノイズの影響で検出が困難になり, 周囲に配慮が必要な場所では使用が難しいといった欠点がある. そこで本研究では, ジェスチャーの検出を行うために各種センサを取り付けた手袋型デバイスであるデータグローブ「GROVE」を開発し, これらの操作手法を補助する方法を提案する. GROVEとは, 成長を意味する「grow up」と, 手袋を意味する「glove」を合わせた造語である. 図1.1はGROVEシステムのロゴである.

図 1.1 GROVEロゴ

GROVEには, 指の曲げを検出するための抵抗式のひずみゲージや, 手の移動・回転を検出するためのIMU（慣性計測ユニット）センサを搭載しており, 昨年度までの研究で既に, 指文字などのハンドジェスチャーの検出を行うことができている. 本年度は抵抗式のひずみゲー

ジが, ユーザーの指の運動を阻害する問題を解決するために, 銀繊維を配合した導電繊維手袋をセンサとし, 指の検出を行う手法を開発・検討した. また, 本研究は研究の一環として, 指文字を入力するインターフェースを開発し, 難聴者が快適に電子機器への文字入力を行うことができる環境の開発を目的としている. これまでの研究で, デバイスの製作, スマートフォン上の専用アプリケーションの作製, デバイスとスマートフォンの連携に成功している. それに伴い, 「あ」「い」「う」「え」「お」の指文字, プレゼンテーションのスライド送り, カメラの撮影ポーズ等のジェスチャー検出に成功している. また, 本年度は新たに導電繊維を用いた指の曲げの検出, 及び当検出手法を用いたGROVEデバイスの製作と, それを用いたロボットハンド

# 第２章　解説

## ２.１　指文字

指文字とは, 片手のみで言語を表現することができる視覚言語の一つである[2]. 50音を表現することができるため, 主に手話の補助として用いられることが多い. また, 手話と違い地域ごとにおけるいわゆる方言が無く, 統一されたジェスチャーを用いるため広く使用されているという特徴がある. 図2.1に50音の内の「あ」「い」「う」「え」「お」の例を示す.

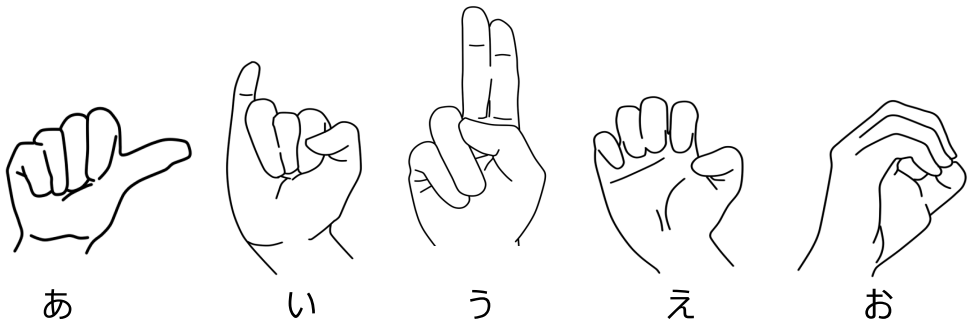


図 2.1 指文字一覧

本研究では, 片手のみの手袋型デバイスで検出できるジェスチャーとして指文字の検出を行った. 2012年度の研究では曲げセンサのみの搭載であったため, 例えば図2.2の指文字の「ぎ」の様な濁点を表現するために, 手を平行に移動させる様な一部の文字の検出を例外としていた.

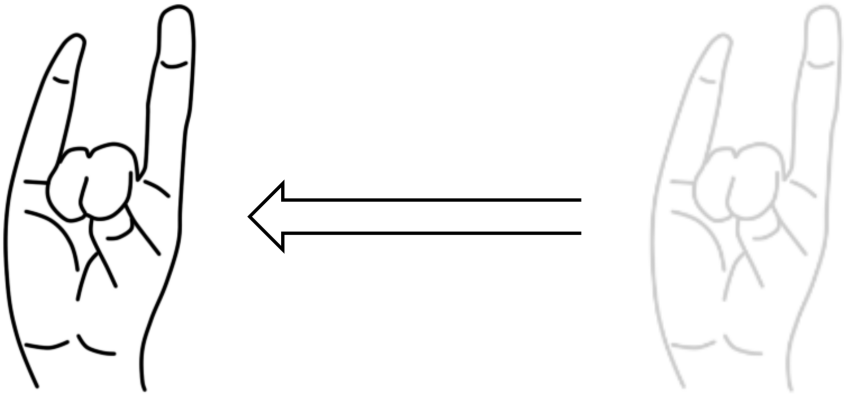

図 2.2 指文字「ぎ」の例

2013年度は慣性計測用センサを新たに搭載することで, 従来では不可能だった手の回転や移動等を検出できるようにした.

## ２.２　ハンドジェスチャーの検出手法

指文字などのハンドジェスチャーを検出する手法として以下の様なものが挙げられる.

1) イメージセンサを用いた手法

   (a) 画像処理を用いた手法

   (b) 3D深度センサを用いた手法

2) 物理的センサを用いた手法

   (a) EMG（筋電位）を用いた手法

   (b) データグローブを用いた手法

また, データグローブを用いた手法において, ハンドジェスチャーを検出するために指の曲げの検出が重要となるが, 検出手法として以下のようなものが挙げられる.

1) 機械的手法

2) 光ファイバを用いた手法

3) ひずみゲージを用いた手法

4) 手形状に沿った特殊なセンサを用いた手法

5) 導電繊維を用いた手法

2012・2013年度の本研究では, コスト・容易性等の条件からひずみゲージを用いた手法を選択したが, 本年度の研究において, 新たに導電繊維を用いた手法を提案し, 新しいデータグローブの開発を行った. 次項よりこれらのジェスチャー検出手法の特徴について述べる.

### ２.２.１　画像処理を用いた手法

この手法ではカメラセンサから送られてきたイメージを基に, 人体の手の動きや指の動きを検出する. 一般的に特徴抽出やパターンマッチングを行い, 手や指がどのような状態にあるかを検出し, それらの時間的変化を元にジェスチャー検出を行う手法である. 一台のカメラのみで検出することができる利点があるが, 精度を上げるのが難しく, 計算処理に時間がかかるといった欠点がある. また, カメラと手の間に一定以上の距離を設ける必要があり, 例えば外出時に用いることを想定すると, カメラを取り付ける位置が問題となり, その解決手法として首からネックストラップを用いて吊り下げて使用する手法が挙げられる[3]. しかし, カメラの視野角に収まるように手を前方に挙げる必要があり, 常に使用するのは難しい.

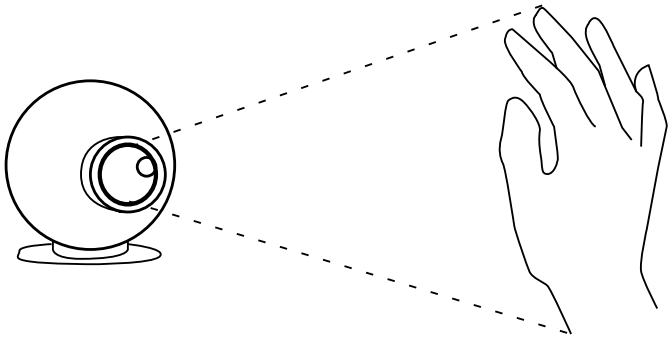

図 2.3 RGBカメラを用いた手法

### ２.２.２　3D深度センサを用いた手法

3D深度センサとは, 単純なRGB入力だけではなくセンサから被写体までの距離も測定するセンサのことである. 3D深度センサとしては, Microsoft社が提供する「Kinect」や, Primesense社が提供する「Xtion」または, SoftKinetic社のTOF形式を用いた「DepthSense」が有名である. 画像処理を用いた手法に比べて, 精度が高いという利点があるが, 一般的に3D深度センサは2台以上のイメージセンサを間隔を開けて配置する必要があるため, 通常のカメラセンサに比べて大型になるが, 肩の上に載せて利用する手法がある[4]. また, コストも高い.

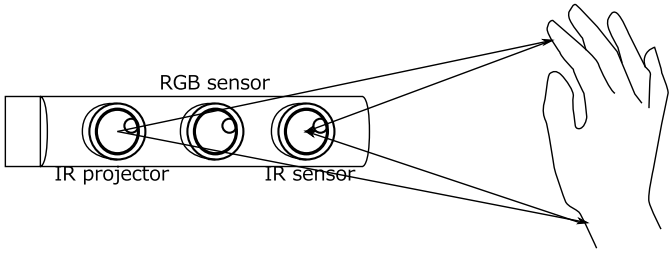


図 2.4 3D深度センサを用いた手法

### ２.２.３　EMGを用いた手法

EMG(ElectroMyoGraph)とは, 人間が筋肉を動かす際に命令を神経に乗せて体に送る時に発生させる微小な電流のことである. 物理的なセンサの中では小さなデバイスで計測が可能だが, 測定制度がノイズに非常に左右され易く測定が難しい. 現在研究が盛んに行われている段階で, 実用化はまだあまりされていない. （米Thalmic社の「Myoアームバンド」等はあるが, 一般販売は現時点ではまだ行なわれていない. ）また, 精度を高くするには高額な機器が必要である.

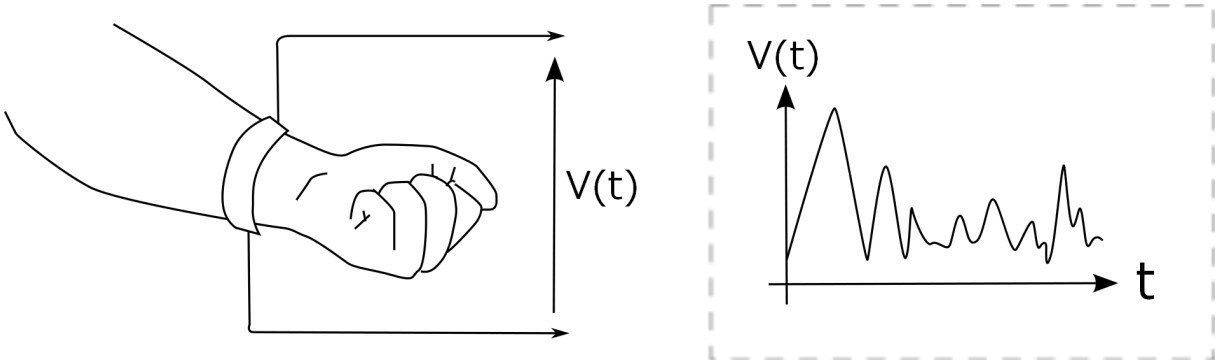


図 2.5 EMGを用いた手法

### 2.2.4　光ファイバーを用いた手法

手袋の指に沿って配した光ファイバの光透過率が, 指を曲げることにより変化することを利用して, ファイバの一端からLED(発光ダイオード)で入力した光を, 他端に配したフォトトランジスタで測定し, その光量から指の曲げ具合を測定するという手法である. この際, 光透過率を任意に設定するために微小なスリットを光ファイバに入れる場合が多い. 下記に挙げる物理的センサと違い, 指の曲げだけでなく, 指の開きの検出まで行うことができるが, 一般的にコストが高い. また, あまり実用化されておらず, センサが市販されていない.

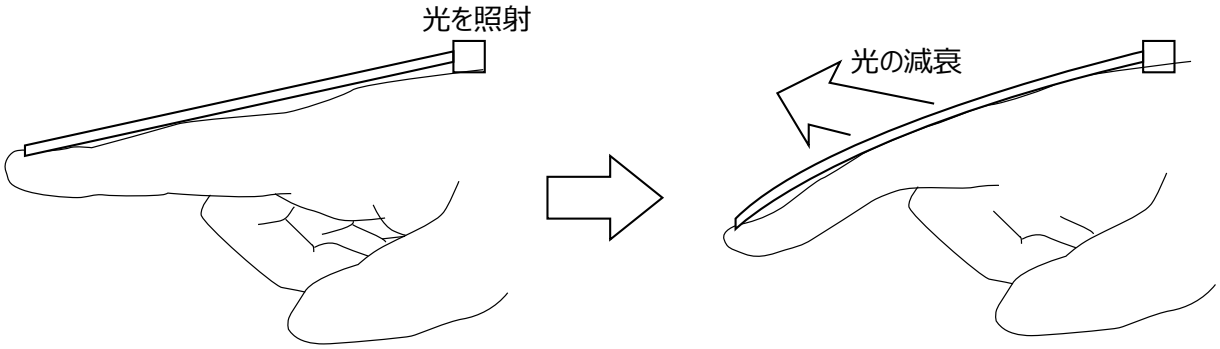


図 2.6 光ファイバーを用いた手法

### 2.2.5　機械的手法

スライド式可変抵抗や, 回転式可変抵抗を用いて指の曲げに応じて, 抵抗値が変化する機構を作成し, それを用いる手法である. 一般的にコストが安くすむ利点があるが, 複雑で大型化しやすい欠点がある.

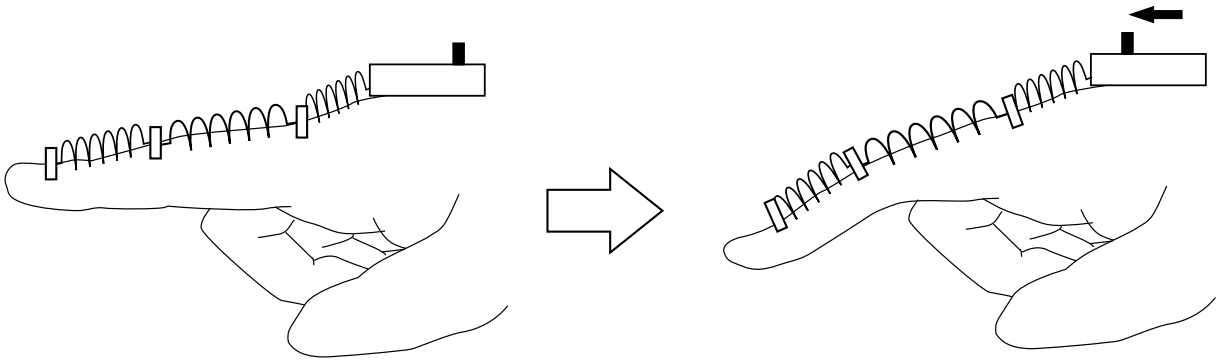

図 2.7 機械的な手法

### 2.2.6　ひずみゲージを用いた手法

手袋の指に沿ってひずみゲージを貼り付け, 指の曲げに応じて変化する抵抗値を読み取る手法である. 一般的に検出が容易で, 比較的安価なため最もよく用いられる手法である. 本

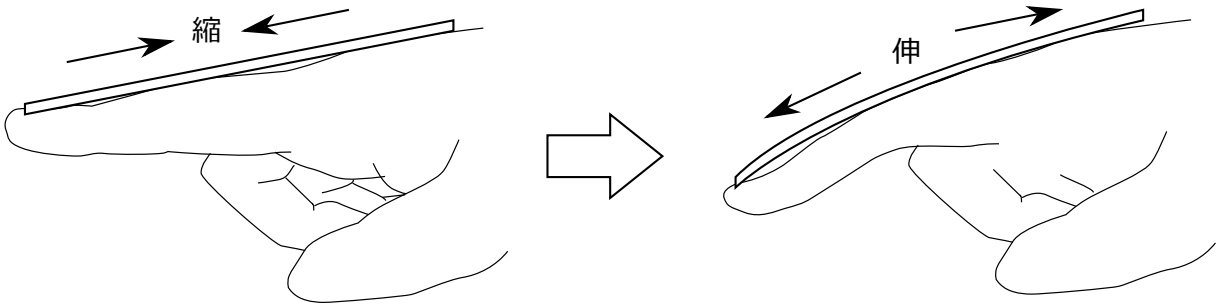


図 2.8 ひずみゲージを用いた手法

研究では外出先で用いることを考え, 人体から一定以上離れた場所に置かなければならないカメラを用いた手法ではなく, 物理的センサを用いて直接指の曲げを検出する方法を選択した. また, 物理的センサを用いた手法の中でも最も扱いが容易なひずみゲージを用いた手法を選択した.

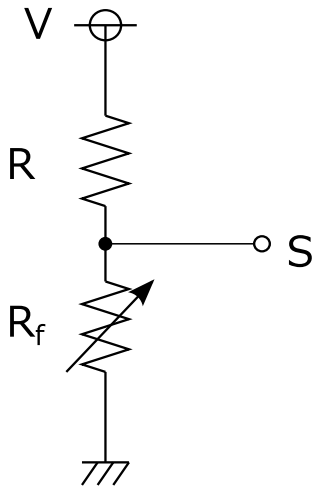


図 2.9 ひずみゲージを用いたセンサ回路

曲げ度の検出には図2.9のような分圧回路を用いる. この時, センサの端子Sには以下のような電圧が生じる

$$V_S = \frac{R_f}{R + R_f} \times V \tag{2.1}$$

この分圧の式からわかるとおり, $R_f$の値が上昇すればSの電圧が上昇する. この電位を計測することで曲げ度を検出する.

### 2.2.7　高密度触覚センサ

フレキシブル基板を手指の形状に沿ったパターン状に加工し, センサとして用いる手法がある. 東京大学, 産業技術総合研究所, 日本メクトロン株式会社が共同で開発している「高密度触覚グローブ」などの様な手勢検出方法などがある[5]. この研究では, 図2.10の様にフレキシブル基板を細かい枝状に成型し, 手の形状に沿って貼り付け, その後ポリウレタンフィルムで全体を覆うことで, 耐久性を保ちながら可動性が高く, 触感性の良いデータグローブの開発に成功している.

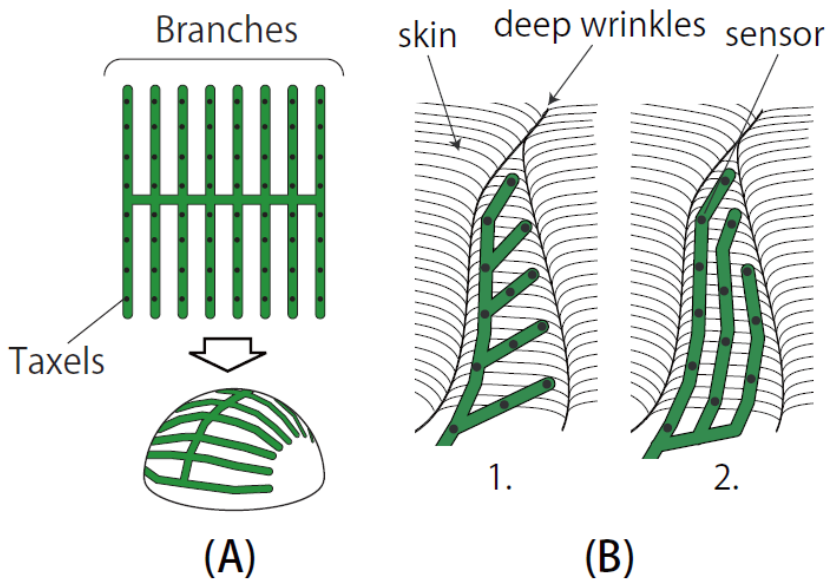


図 2.10 (A)枝状触覚センサによる曲面実装. (B)枝状センサシートの枝の配置法.
1.短い枝を多数配置した場合. 2.近隣の長いシワに沿った長い枝を少数配置した場合.
(出典 [5])

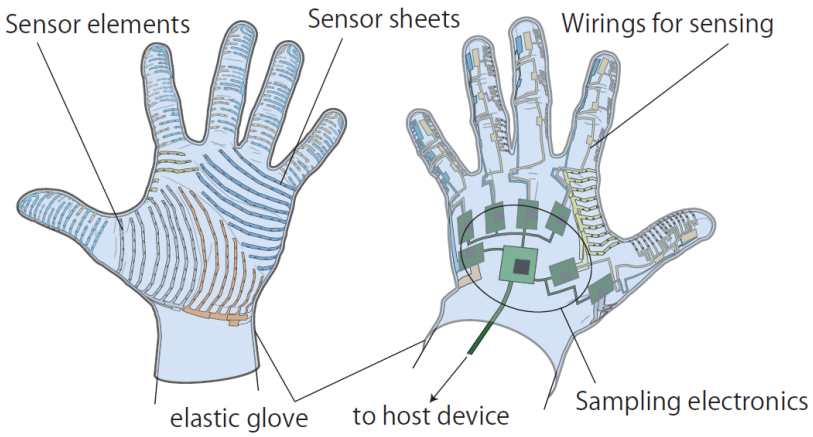


図 2.11 高密度触覚グローブ(出典 [5])

### 2.2.8　導電繊維を用いた手法

前項までで紹介したセンサ等を手袋に取り付ける手法では, ユーザーの指の曲げ運動を阻害したり, センサを新たに搭載する必要があるため, コストが高いといった問題があった. また, 高密度触覚グローブではこれらの問題は解決されているものの, 製作難度が高いといった問題があった. そこで, 本研究では図2.12銀繊維を配合した導電繊維を編み込んで製作された手袋をセンサとして用いる手法を考案した. この導電繊維は若干の抵抗を持っており,

図 2.12 本研究で用いた導電繊維編込み手袋(Agloves for Touch screen)

指を曲げにより繊維が伸びることによる抵抗値の変化と, 指を曲げた時に導電繊維同士が接触することで, センサ端子間距離が短くなり, 抵抗値が低くなる現象を用いて曲げの検出を行う.

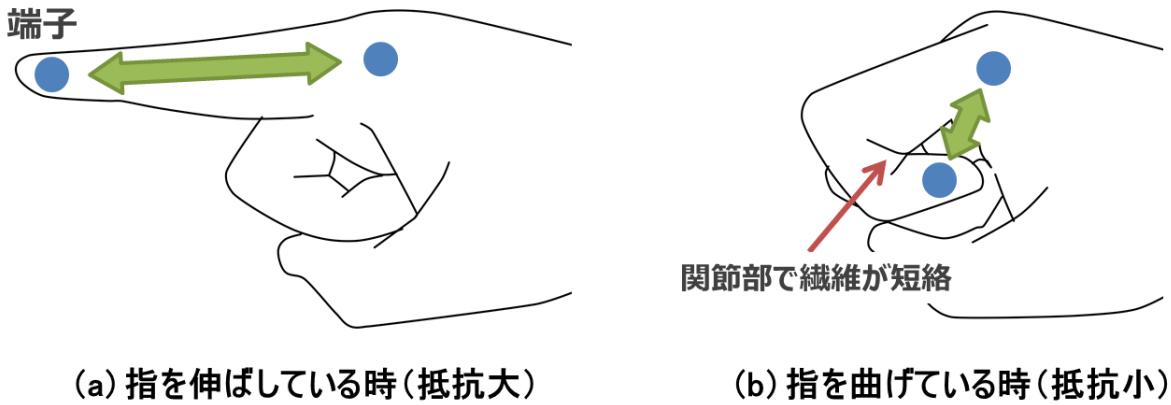


図 2.13 導電繊維を用いた曲げ検出の原理図

### ２.２.９　検出方法の比較

表2.1に, 今回比較検討した指の曲げを検出する手法について示す.

表 2.1 指の曲げを検出する手法の比較

| 検出手法 | 外出時における設置の容易性 | コスト |
|---|---|---|
| 画像処理 | 難 | 高 |
| 3D深度センサ | 難 | 高 |
| EMG | 易 | 高 |
| 機械的手法 | 中(大型化しやすい) | 低 |
| 光ファイバ | 易 | 高 |
| ひずみゲージ | 易 | 低(導電繊維よりは高い) |
| 高密度触覚センサ | 易 | 高(製作難度が高い) |
| 導電繊維 | 易 | 低 |

本研究では外出先で用いることを考え, 人体から一定以上離れた場所に置かなければならないカメラを用いた手法ではなく, 物理的センサを用いて直接指の曲げを検出する方法を選択した. 2012・2013年度は, 物理的センサを用いた手法の中でも最も扱いが容易でコストが低いひずみゲージを用いた手法を選択した. 2014年度は, ひずみゲージよりも触感性が良く, 製作コストも低くなる導電繊維を用いた検出手法を選択した.

# 第３章　システムの構成

## ３.１　GROVEシステムのコンセプト

本研究では, タッチパネルや音声操作等に代わるスマートフォンを操作する手法として, データグローブを用いてハンドジェスチャーを検出し, 検出したジェスチャーを用いて操作を行う手法を提案する. 本手法で提案するシステムのこと「GROVEシステム」とし, システムの一環として作成したデータグローブを「GROVE」と命名した. また, 「GROVE」とは成長を意味する英語の“ grow up ”と, 手袋を意味する“ glove ”から成る合成語である.

GROVEシステムは外出先で歩行時などに, GROVEデバイスを用いて任意のハンドジェスチャーを検出し, スマートフォンを用いて電話を掛けたり, 写真を撮ったりするシステムである. システムの構成は図3.1に示すとおりである. ポケットや鞄にスマートフォンを収納した状態でも運用することができるよう, スマートフォンとデバイスの接続はBluetoothを用いた無線通信を行う様にした. 手の移動・回転を検出するためにIMU(慣性計測用ユニット)センサを搭載しており, 加速度と角加速度を検出することができる. 2012・2013年度の研究において, 指の曲げの検出には抵抗式のひずみゲージを5指分搭載していたが, 今年度の研究において, 導電繊維編込み手袋を用いて曲げ検出用センサを構成した.

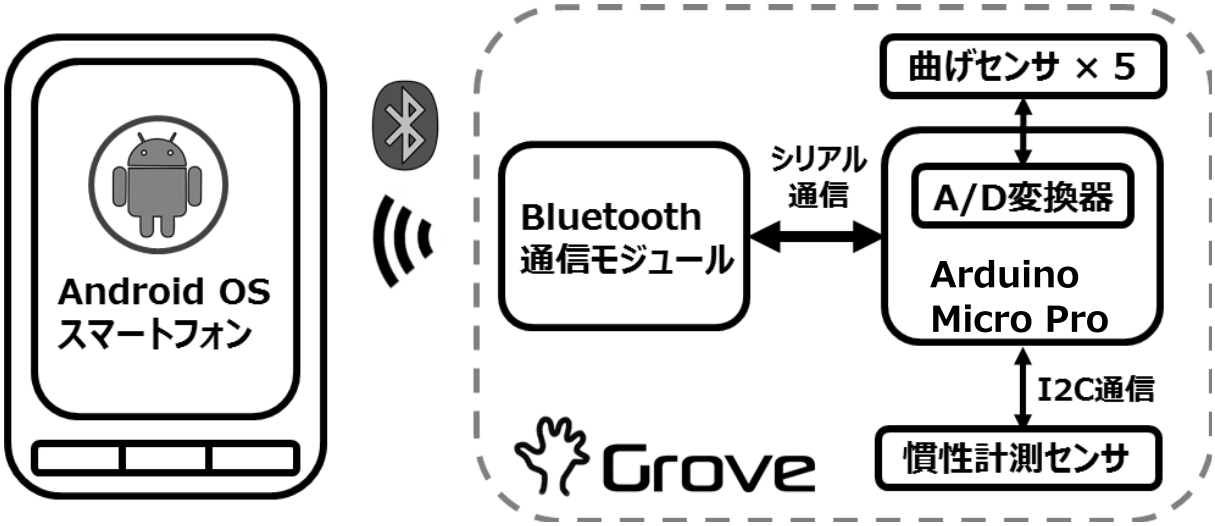


図 3.1 GROVEシステムの構成

## ３.２　ハードウェアの構成

### ３.２.１　全体の構成

図3.2に, ハードウェア全体の構成を示す.

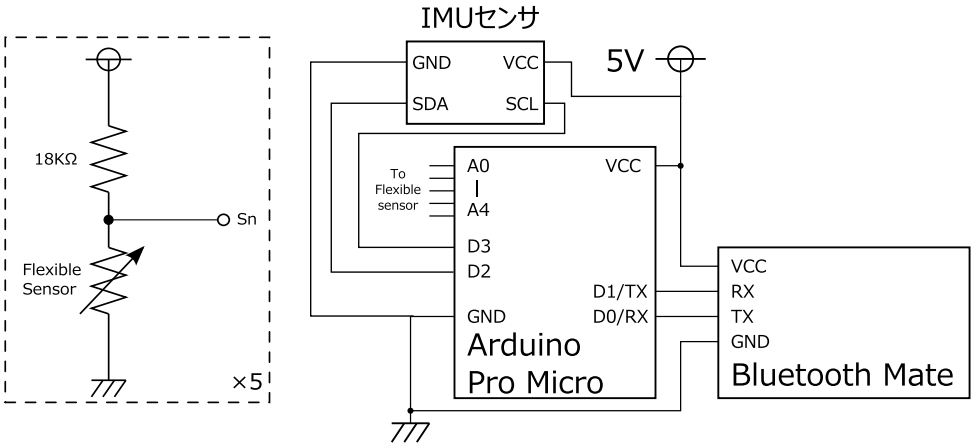


図 3.2 ハードウェアの構成

ハードウェアを構成する各ブロックの詳細について以下に述べる.

1) 曲げセンサ部

2012・2013年度の構成では, 可変抵抗であるひずみゲージと抵抗の分圧器になっており, ひずみゲージの抵抗値に比例して端子電圧が変化する. 本年度の構成では, ひずみゲージを導電繊維を用いたセンサに置き換えて構成している.

2) IMU(慣性計測ユニット)センサ

2013年度版以降のもので採用. これにより手の移動・回転の検出を必要とするジェスチャーの検出が可能になっている.

3) Arduino Pro Micro

2012年度版のGROVEではArduino Unoを採用している. ひずみゲージ部の端子電圧をAD変換しており, このとき, 端子電圧0-5Vを0-1023のデジタルデータに変換している. IMUセンサとはI2C通信, BluetoothMateとはシリアル接続している.

4) BluetoothMate

内部にBluetooth通信モジュールが入っており, 固有アドレスも割り当てられているため, AndroidOSスマートフォンはこの素子と同期を行うことで, Arduinoからデータを受信する(SPPプロファイルを使用)ことができるようになっている.

次項より, 本研究において各年度ごとに実際に作成したGROVEデバイスの構成について述べる.

### ３.２.２　2012年度版

図3.3に2012年度に作成したGROVEデバイスの写真を, 図3.4にセンサ等の素子類の配置図を示す.

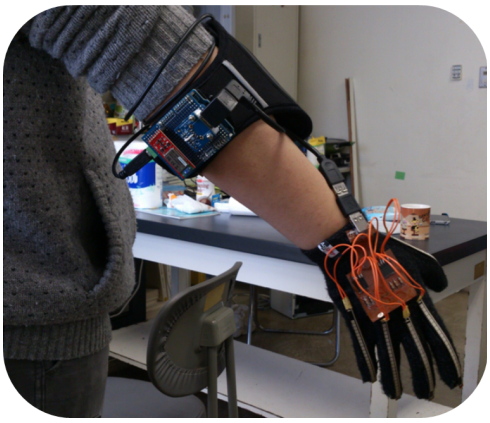

図 3.3 2012年度版GROVEの写真

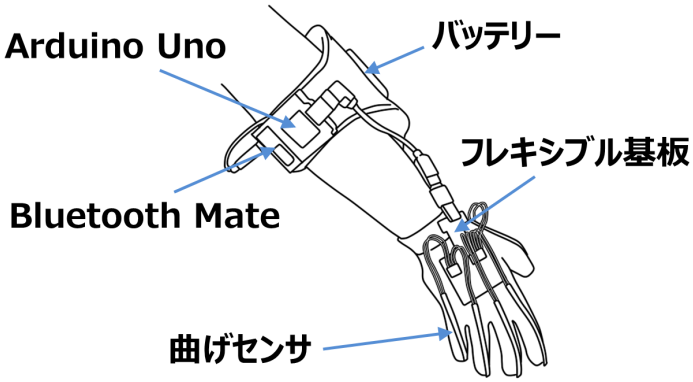


図 3.4 2012年度版GROVEの素子構成

2012年度版のデバイスの重量は約400g程度であり, バッテリーや回路等はアームバンド内に収納する様な構造になっていた. 以下に, 2012年度版の主な特徴について述べる.

- 曲げ検出にはコスト・構築の容易性等から抵抗式のひずみゲージを採用
- 指の曲げ検出用の抵抗の分圧回路を手の甲上のフレキシブル基板に実装
- デバイスの各種計算･通信を行うためのプロセッサとしてマイコンボードであるArduino Unoを採用
- Bluetooth通信を行うためにBluetooth MateをArduinoの拡張基板であるシールド上に実装
- 手の甲上の基板と, Arduino間の接続にはUSB3.0規格のコネクタを利用
- バッテリーとして取扱いの容易性から市販のモバイルバッテリーを利用

手の甲上の分圧回路の実装にフレキシブル基板を用いたのは, 手指を動かした際に, 通常の基板だと剛性の都合上, 運動を阻害する恐れがあったためである. しかし, 基板の薄さから断線等が起きやすく, 次年度以降のモデルには採用していない. メインのプロセッサにはマイコンボードであるArduino Unoを用いた（マイコンはATMEL社製ATmega328p）. これは, 開発の容易性等が主な理由であり, 実際のジェスチャーの検出にかかる複雑な処理は, 全てスマートフォン上で行うことを前提とした設計である.

GROVEとArduino Unoとの接続にUSB3.0コネクタを利用した理由を以下に述べる.

- 広く一般に普及している規格のため, コネクタ・ケーブル共に入手が容易
- 回路の設計上7ピン必要だったが, 9ピンまで対応しているため十分である
- コネクタが細く薄いため, 手首の曲げ等に干渉しにくい
- 2.54ミリピッチへの変換コネクタが存在する

安価で入手しやすく, 取り回しの効きやすい規格であるため採用したが, 当初の予定ではUSB3.0microBコネクタを利用する予定であったが, microBのメスプラグが基本的に民生用に流通が少なく, 入手が困難であったため断念した.
また, USB3.0コネクタの各ピンの割り当ては以下のようになっている.

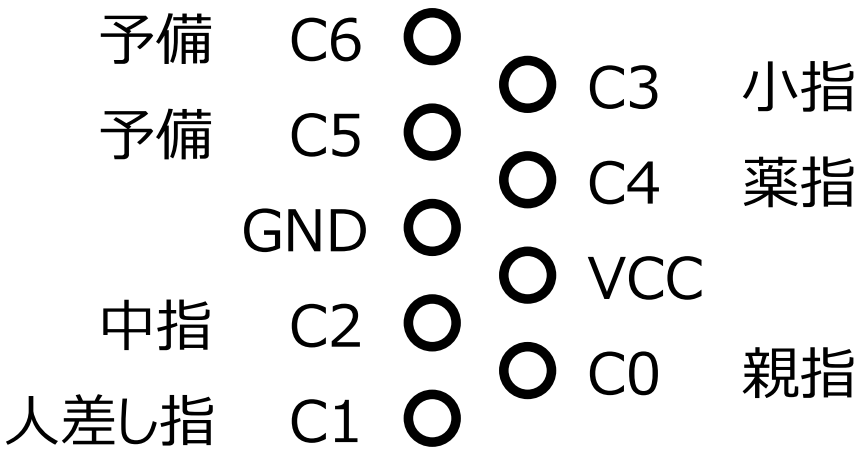


図 3.5 USB3.0コネクタのピン割り当て

これは, 手袋上のひずみゲージ回路パターン図の作成に合わせて規格した並びである. 今後のセンサの増設などの拡張性を考慮し, 2ピンの余りを残した. 本来であれば, USB3.0コネクタは7ピン＋GND用2ピン（基板上で導通させて用いる）を想定して策定された規格であり, デバイスの作成時に用いたコネクタのピッチ変換基板でも, 2つのGND用ピンが導通するように設計されていたが, これを加工し, 独立したピンとして利用している.

### ３.２.３　2013年度版

図3.6に2013年度に作成したGROVEデバイスの写真を, 図3.7にセンサ等の素子類の配置図を示す.

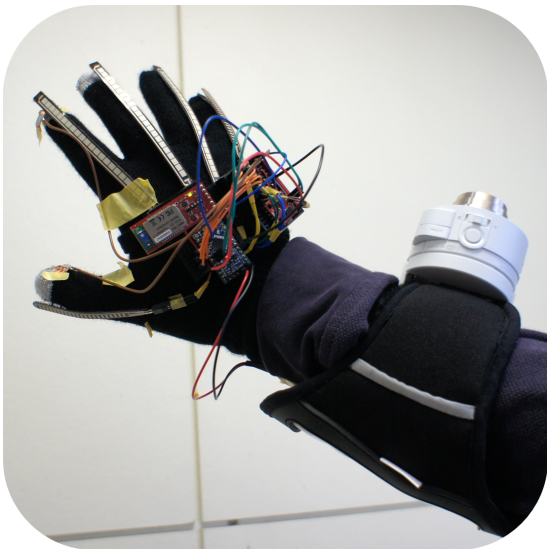

図 3.6 2013年度版GROVEの写真

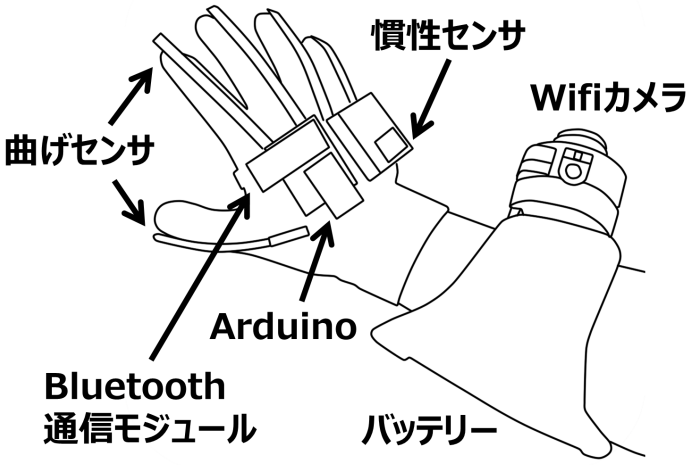


図 3.7 2013年度版GROVEの素子構成

2013年度版GROVEを2012年度版と比較した際の特徴を以下に述べる.

- WiFiカメラ（SONY社製DSC-QX10）を搭載
- 親指と人差し指の先に導電繊維を編み込んだ電極を作成
- 回路類をすべて手の甲上に配置
- バッテリーを市販のモバイルバッテリーから, 素のリチウムイオンポリマーバッテリー(1000mAh)に変更
- IMUセンサを搭載したため, 手の移動・回転等の検出が可能になった
- 回路は全てブレッドボード上で構成
- 全体に導電繊維が編み込まれた手袋から, 指先のみ導電繊維が編み込まれたものに変更
- 小指のひずみゲージのみ小型の物に変更

WiFiカメラを搭載し, スマートフォンとWiFi接続することで, 写真撮影ジェスチャーをスマートフォン上で検出し, スマートフォンからWiFi経由でカメラの操作と, 撮影した写真画像の取得を行うことが可能になった.

2012年度版のモデルで, 手袋側とアームバンド側に回路を分散し, 間をUSB3.0ケーブルで接続していたため, 重量・装着感等で問題となっていた. そのため, 2013年度版では手の甲上の回路のみで完結するように変更した. また, 軽量化を図るために汎用性の高いモバイルバッテリー（SANYO製eneloop mobile booster）から, リチウムイオンポリマーバッテリーに変更した.

2012年度版では指の曲げのみしか検出できないため, 図2.2の様な手の移動・回転の検出が必要なハンドジェスチャーの検出を行うことができないといった問題があったが, 2013年度版のモデルではIMUセンサを搭載することでこれを解決した.

回路の実装にはブレッドボードを用いている. これは, 回路の修正等の容易性のためである. 当初の予定ではガラス基板上に実装する予定であり実際に作成したが（図3.8）, 回路基板の設計ミスにより, エッチング作業後に基板の不備や, 複数枚の基板を互いに組み合わせることができず断念した.

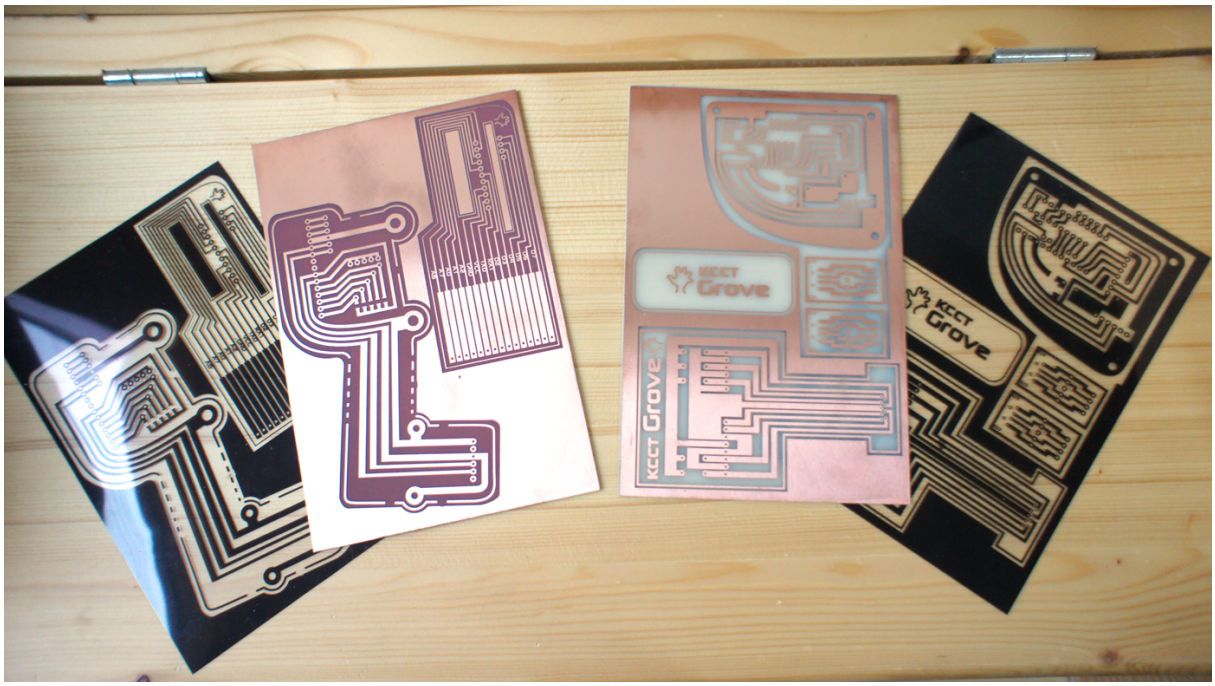


図 3.8 実際に作成した基板

### ３.２.４　2014年度版

図3.9に2014年度に作成したGROVEデバイスの写真を, 図3.10にセンサ等の素子類の配置図を示す.

図 3.9 2014年度版GROVEの写真

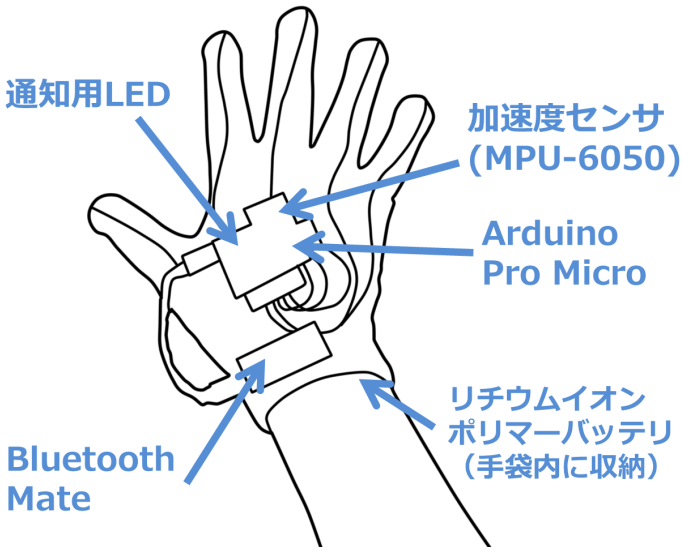


図 3.10 2014年度版GROVEの素子構成

2014年度版GROVEの特徴を以下に述べる.

1) 指の曲げの検出に, 抵抗式のひずみゲージではなく導電繊維を編み込んだ手袋自体を用いている.

2) 各種状態の通知用LEDを搭載

3) IMUセンサを安価なもの（MPU-6050）に変更

4) 回路を2.5cm角のコンパクトなユニバーサル基板上に実装した

2014年度版のデバイスでは, 手袋に編込まれた導電繊維を直接センサとして用いることで軽量かつ装着性の良さの向上を図った. 2012年度・2013年度は, デバイスの重量が約400g（2013年度版はカメラ重量を含む）だったのに対し, 約65gと大幅な軽量化に成功している. さらに, デバイス一つ当たりのコストの削減にも成功しており, 2012・2013年度版が共に約26000円程度（2013年度版はカメラ代含まず）だったのに対し, 2014年度版は半分以下の約12400円程度に収まっている.

また, 従来のデバイスではスマートフォンをポケットに収納している場合, 手袋単体ではユーザーへのなにかしらの出力装置が無かったが, 2014年度版では通知用に2つのLEDを搭載することでこれを解決している.

### ３.２.５　各年度の比較

以下の表3.1に, 各年度ごとのGROVEデバイスの比較を示す.

表 3.1 指の曲げを検出する手法の比較

| | 2012年度版 | 2013年度 | 2014年度版 |
|---|---|---|---|
| マイコン | Arduino Uno | Arduino Pro Micro | Arduino Pro Micro |
| 重量 | 約400g | 約400g(カメラ込) | 約65g |
| 曲げセンサ | 抵抗式ひずみゲージ | | 導電繊維 |
| 慣性センサ | なし | IMU SDOF | MPU-6050 |
| カメラ | なし | SONY DSC-QX10 | なし |
| 通信 | Bluetooth2.1(Bluetooth Mate) | | |
| 通知用LED | なし | | あり |
| 手袋 | 全体導電繊維編込 | 指先のみ導電繊維 | 全体導電繊維編込 |
| バッテリー | SANYO社製 eneloop mobile booster | | Li-ionバッテリー(1000mAh) |
| コスト(概算) | 26500円 | 26000円(カメラ別) | 12400円 |

上記の通り, 2012年度から2013年度にかけては機能が多く追加されており, 手の回転・移動の検出や, 無線での写真撮影等が可能になっている. 一方, 2014年度版では約65gと大幅な軽量化に成功しており, 低コスト化にも成功しているが, 2013年度版で搭載されていた機能をいくつか削減している. これは, 2013年度版がGROVEデバイスの有用性を試みるためのバージョンであったのに対し, 2014年度版では実用性の向上を特に重視したためである. そのため, 精度が多少劣るものの安価な慣性センサを搭載したり, バッテリーの容量が少ないがコンパクトなものに変更したりなどの改良を図っている.

## 3.3　ソフトウェアの構成

### 3.3.1　全体の構成

以下にソフトウェア全体の構成図を示す.

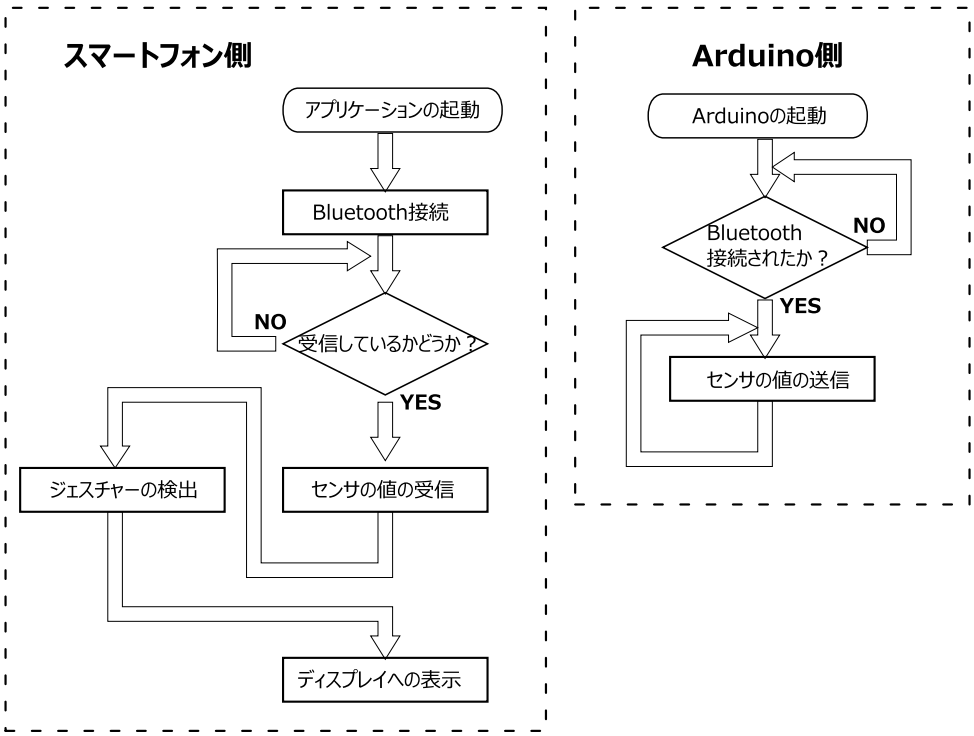


図 3.11 ソフトウェア全体の流れ

順を追って, 全体の流れについて以下に説明する.

1) スマートフォン側でアプリケーションを起動する.

2) Arduinoに電源を接続し, 給電を開始する.

3) スマートフォン側からArduino側へBluetooth接続設定を行う.

4) Arduino側は接続設定終了後はセンサの値を読み取り, スマートフォンへと送信し続ける.

5) スマートフォン側は受け取ったデータを元に, ジェスチャーの検出を行う.

以上の手順で, ジェスチャーの検出を行う.

主にジェスチャー検出のためにデータを元に計算を行うのはスマートフォン側で, Arduinoはデータの送信用に用いている.

### ３.３.２　Arduino側

以下にArduino側のソフトウェアの流れを示す.

今回, 本研究ではArduinoはあくまで, ひずみゲージの出力値のAD変換, 慣性計測用センサの

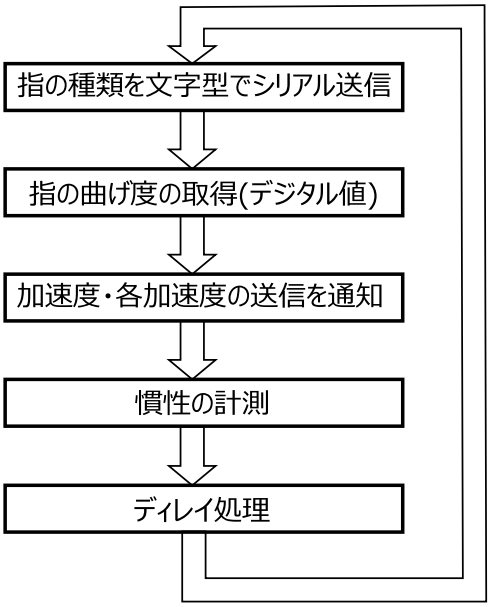


図 3.12 Arduino側のソフトウェアの流れ

データの取得, Bluetooth通信用にのみ用いているため, 簡素な流れとなっている. AD変換等はArduino側で自動的に行うため, ソフトウェア側での制御は不要である. ひずみゲージの分圧回路によって出力値の0-5Vを0-1023にAD変換している. また, 慣性計測用センサとArduinoはI2C通信でデータのやり取りをしている. なお, Bluetooth通信を実際に行うとスマートフォンが受信中である際に, Arduinoが続きを送信しようとしてデータが崩れてしまうことがあるため, 30mSほどのディレイ処理を挟んである.

### ３.３.３　スマートフォン側

以下にスマートフォン側のソフトウェアの流れを示す.

ソフトウェア起動から, ジェスチャーを検出するまでの流れは図3.14の通りである.

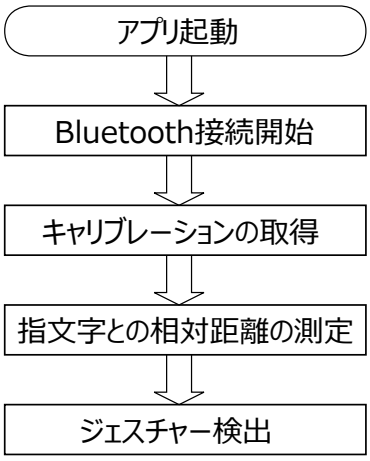


図 3.13 スマートフォン側のソフトウェアの流れ

以下に手順を示し, 解説する.

1) Bluetooth接続

Bluetooth接続では, BluetoothモジュールのMACアドレスを指定して接続を行う.

2) キャリブレーションの取得

キャリブレーションの取得では, 指の曲げによる曲げ値の最大値と最小値の設定を行う, また, 実際に指を曲げジェスチャーをとった際の各指の曲げの値を設定する必要がある.

3) 指文字との相対距離の測定

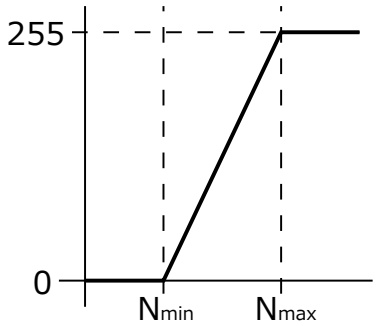


図 3.14 指の曲げ度変換フィルター

ここでは, キャリブレーションの取得で得られた最大値・最小値を元に線形フィルタリングを用いて曲げの値を0-255に変換する.

これは, GROVEをつけた人によって変化する指の曲げ具合に対応させるためである.

$$D_{out} = \frac{255}{N_{max} - N_{min}} \times (D_{in} - N_{min}) \tag{3.1}$$

実際には, 式3.1の計算式を用いている.

4) 慣性計測

慣性計測用センサとI2C通信を用いて, 加速度と各加速度を検出する. なおこの際, センサの仕様上, 定期的にデータの初期化をする必要がある. これは, 加速度等の信号に誤差があるため, それを積分して得られるデータのずれが常に広がるためである.

5) ジェスチャー検出

キャリブレーションの取得で得られた, ジェスチャーを取った際の各指の曲げ度と, 現在の曲げ度との距離を各指の曲げ度の差分の二乗和を平方根することによって得る.

$$\text{距離} = \sqrt{\sum_{N=\text{親指}}^{\text{小指}} (\text{曲げ値}_N - \text{ジェスチャーの設定値}_N)} \tag{3.2}$$

この時の距離が一定値以下ならジェスチャーの検出を行う.

また, 一部の指文字においてはこの曲げ値から求まる距離以外にも, 現在デバイスが動いているか, 振られているか等を慣性計測用センサで取得した値から求めて, 検出条件として用いる場合も存在する.

# 第４章　まとめ・結果

## ４.１　ジェスチャーの検出

### ４.１.１　GROVEシステムビューの解説

以下にGROVEシステムのスマートフォン側のUIを示す.

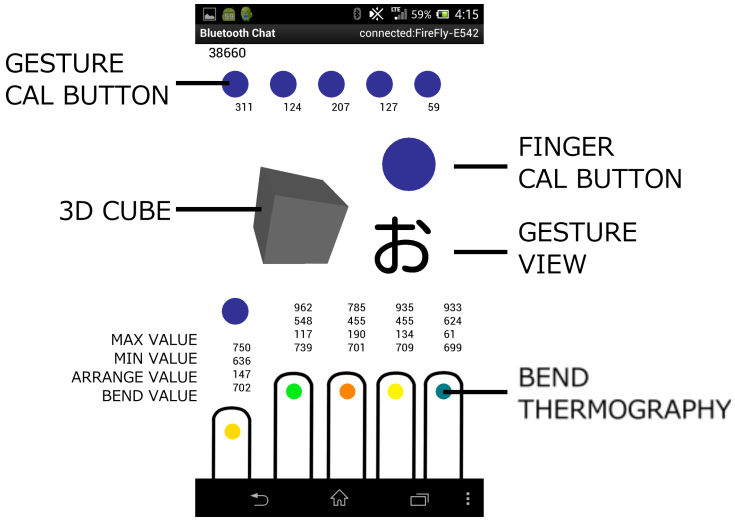


図 4.1 GROVEビューの解説

以下にGROVEビューの各値やボタンについての説明を述べる.

- GESTURE CAL BUTTON

  ジェスチャーをとった際の, 各指の曲げ値を保存するのに用いる. キャリブレーションの取得が終了した後ジェスチャーをとり, 曲げ値が安定してからボタンを押す必要がある.

- FINGER CAL BUTTON

  指の曲げ値の最大値と最小値を得るためのキャリブレーションボタンである. このボタンを押してから７秒間ボタンが緑色に変化するため, その間に手で, 握る・開くを繰り返し, 指の曲げ値の最大値と最小値を記憶させる.

- GESTURE VIEW

  現在, 検出されているジェスチャーを表示する.

- 3D CUBE

現在, どの程度手が握られているか, 回転しているかを3Dデータを用いて, 視覚的にわかりやすく表示している.

- VALUE
  - MAX VALUE

    FINGER CAL BUTTONを押して取得した際の指の曲げの最大値を表示する.
  - MIN VALUE

    FINGER CAL BUTTONを押して取得した際の指の曲げの最小値を表示する.
  - ARRANGE VALUE

    指の曲げ値を0-1023から0-255に線形フィルタリングした後のデータを表示する.
  - BEND VALUE

    Arduinoから送られてきた指の曲げ値をそのまま表示する.
- BEND THERMOGRAPHY

  指の曲げ度(ARRANGE VALUE)のデータを元に, 曲げの強度に応じた色を表示する. 表示される色は図4.2のとおりである.

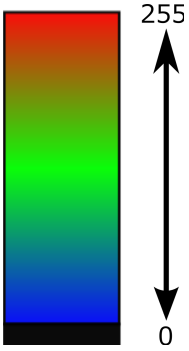


図 4.2 曲げ度に応じたサーモ色一覧

#### ４.１.２　GROVEへのBluetooth接続

以下にGROVEシステムのスマートフォン側のソフトウェアの操作手順を図4.3-4.5示す.

GROVEシステムでは初めにGROVEへのBluetooth接続の設定を行う必要がある. この設定を行うと, Arduinoから手袋の曲げの値が送信される様になる. ここでは, アプリ起動後にまずメニューボダンを押してから, 接続するデバイス一覧を表示し, Bluetooth通信モジュールである「FireFly」を選択する必要がある.

初回接続時のみパスワードの入力が必要だが, 2回目以降は自動で接続が行われる.

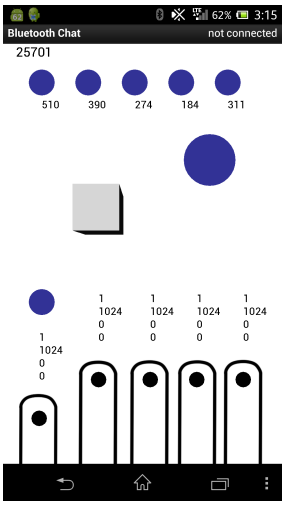


図 4.3 操作手順1

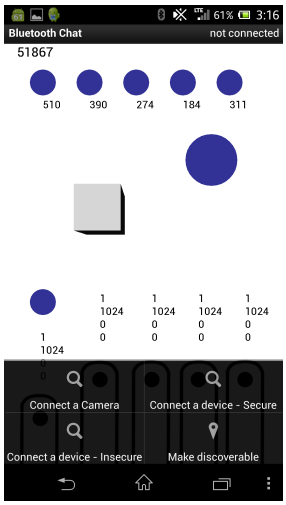


図 4.4 操作手順2

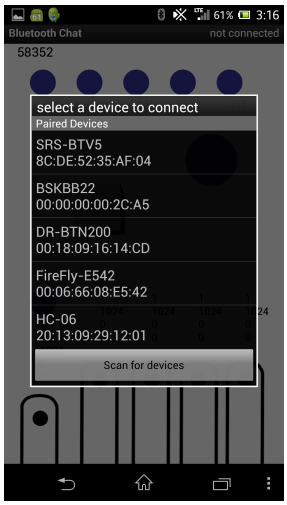


図 4.5 操作手順3

### 4.1.3 Wi-Fiカメラとの接続

以下にWiFiカメラ（ソニー製DSC-QX10）へWiFi接続する際の, GROVEシステムのスマートフォン側のソフトウェアの操作手順を図4.6-4.8示す.

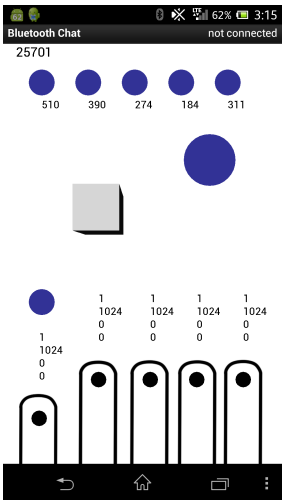


図 4.6 操作手順1

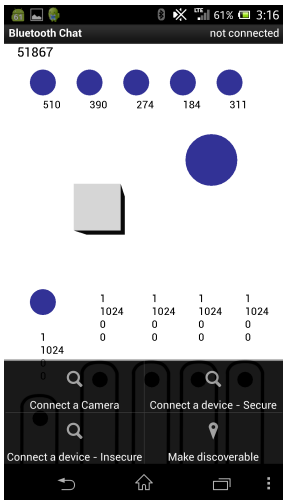


図 4.7 操作手順2

図 4.8 操作手順3

GROVEシステムではWiFiカメラと連動したアプリケーション制御を行うにあたって, まずカメラとの接続をする必要がある.

ここでは, アプリ起動後にまずメニューボダンを押してから, カメラ接続設定ボタンを押し, 周辺のWiFi環境を検索した後, カメラを選択することでカメラとの接続が確立される.

### 4.1.4　ジェスチャーの検出

図4.9にジェスチャー検出を行っている際のスマートフォン上の画面UIを示す. Bluetooth接続後は, ひずみゲージの曲げ度の値がArduinoから送信されるため, その情報を元に, まずキャリブレーションの取得を行い, その後ジェスチャーの検出が開始する.

1) キャリブレーションの取得

FINGER CAL BUTTONを押した瞬間から 7 秒間がキャリブレーション取得時間になるため, その間に数回GROVEを手につけたまま握ったり開いたりを繰り返す. すると自動的に最大値と最小値が更新される.

2) 各指の曲げ値の設定

GESTURE CAL BUTTONを押した瞬間, その際の各指のARRANGE VALUEの値が, ボタンに記憶されるため, ジェスチャーをとり, 値が安定してからボタンを押す. また, デフォルトでは各ボタンは左から「あ」「い」「う」「え」「お」に設定されている.

3) ジェスチャーの検出

上記の設定が終了した後から, 正確にジェスチャーが検出される様になるため, 設定したジェスチャーを取ると, 目的の文字が表示されることを確認する.

図4.10に実際のジェスチャー検出の様子を示す.

図 4.9 ジェスチャー検出UI

図 4.10 実際のジェスチャー検出の様子

### 4.1.5　測定精度

GROVEシステムの指文字検出の際の測定にかかる時間や, 精度について検証した. なお, この検証は2013度のものである. 検証方法を以下に示す.

1) ソフトウェア側からしきい値の設定を行った.
2) キャリブレーションの取得を行った.
3) その後, 指文字を表現した際の各指の曲げ値の設定を行った.
4) 手をパーの状態に開いて, 各指の曲げ値が安定するまで待機した.
5) その後評価用に設定したボタンを押し, 指文字を表現した.
6) その後, 指文字の判別完了時点でボタンプッシュ時からの経過時間を記録した.
   検出された指文字が誤っている場合は×と記載し, 誤検出した文字を記録した.
7) (5)-(6)を各指文字(「あ」,「い」,「う」,「え」,「お」)についてそれぞれ5回ずつ行った.

上記の手順をしきい値70と100の場合で, それぞれ(2)-(7)を2回ずつ行った.

なお, しきい値はジェスチャーと完全一致で0, 最大で実測300程度の範囲を取る値の中から設定した値である. 結果を以下に示す. 実際に採ったデータは付録に添付する.

表 4.1 GROVEシステムの検出精度

| | しきい値 70 | | しきい値 100 | |
|---|---|---|---|---|
| | 正解率 | 応答時間[秒] | 正解率 | 応答時間[秒] |
| あ | 100% | 2.14 | 100% | 1.55 |
| い | 100% | 3.04 | 100% | 0.68 |
| う | 100% | 1.30 | 100% | 1.09 |
| え | 90% | 2.46 | 90% | 1.21 |
| お | 60% | 2.07 | 50% | 1.43 |

データを見ると,「お」の検出精度が著しく低いことが分かる. これは, 図4.11の通り, 手を開いた状態から「お」の手を閉じた状態に状態が遷移する過程で間違って「え」が検出されるためである. これを解決する手段として, 指の曲げ度の変化を微分することで傾きを検出し, 傾きが安定した所のみでジェスチャーの検出を行う等が考えられる.

図 4.11 指文字「お」の誤検出の仕組み

## ４.２　ロボットハンドの操作

以下の図4.12に示すとおり, 3Dモデリングソフト「Shade 3D」を用いてロボットハンドの設計を行い, XYZプリンティング社製3Dプリンタ「daVinci 1.0」を用いて印刷し, 作製した.

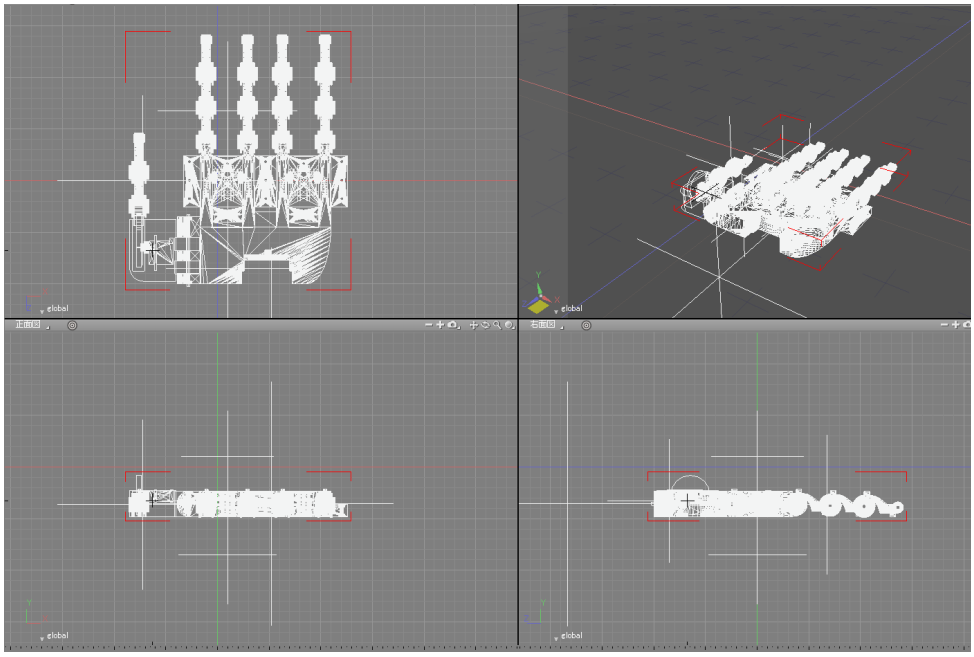

図 4.12 ロボットハンドの設計図

作製したデバイスと, 2014年度版のGROVEデバイスをBluetooth通信で接続し, 図4.13の通り, 無線で操作できることを確認した.

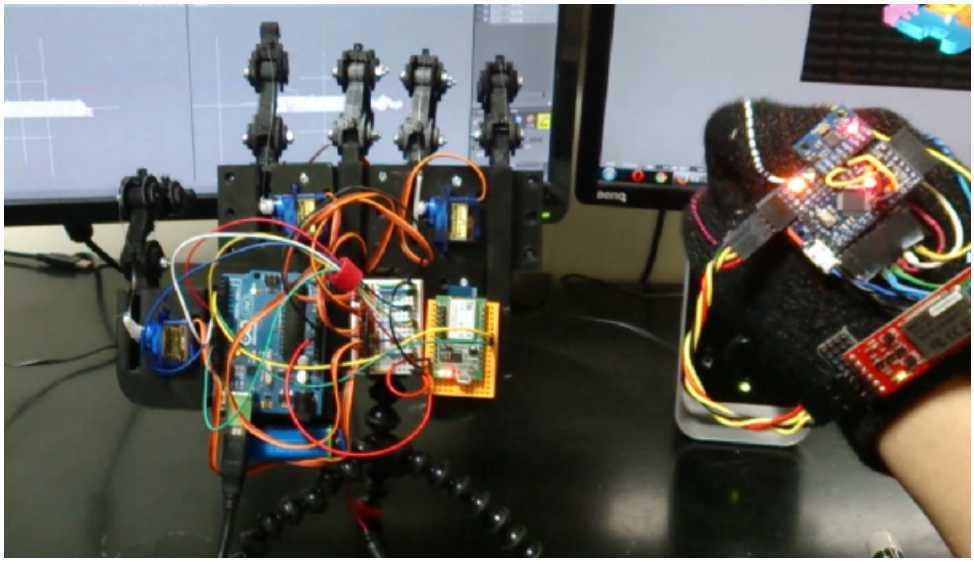

図 4.13 GROVEを用いてロボットハンドを操作する様子

# 第５章　考察

以下に, 本研究で達成したことを述べる.

- 指文字の検出

  抵抗式のひずみゲージを用いて指の曲げを検出し, Bluetooth通信を用いてスマートフォンにデータを送り. 送られたデータを基にユーグリッド距離を用いて,「あ」「い」「う」「え」「お」の指文字や, カメラ撮影のポーズなどのハンドジェスチャーを検出することに成功した.

- 導電繊維を用いた曲げ検出

  抵抗式のひずみゲージを用いた指の曲げの検出は精度が良く, 製作が容易であるといった利点があったが, 実際に作成した結果, コストが高い, 装着感が悪いといった問題があった. 本年度は導電繊維を編み込んだ手袋をセンサとして用いて指の曲げを検出することに成功した. これにより, 従来より安価かつ装着感が良いデバイスの製作に成功した.

- カメラ撮影

  GROVEデバイスとスマートフォンで検出したハンドジェスチャーを基に, Wifi経由でカメラに命令を送り写真を撮影し, スマートフォンに写真データを送ることに成功した. これにより, ユーザーは歩行時にカメラ撮影のジェスチャーをとるだけで写真撮影を行うことができる.

- ロボットハンドの製作と操作

  3Dプリンタを用いて, サーボモーターで各指を曲げ伸ばしすることができるロボットハンドを実際に製作し, Bluetooth通信でGROVEデバイスと接続し, 実際に指の曲げに合わせた操作を行うことができた.

以下に, 今後の研究で引き継ぎ行っていくことを述べる.

- 検出精度の向上

  現在, 表4.1の様に, 一部指文字での検出精度が低いといった問題がある. 今後はユーグリッド距離を用いた手法以外の, K近傍法, 機械学習法, ニューラルネットワーク法等のジェスチャー検出アルゴリズムを用いることで, 検出精度を高めていく必要がある.

- 検出範囲の拡大

  「あ」「い」「う」「え」「お」や, カメラのポーズ等の一部ハンドジェスチャーの判別には成功したが, さらに全ての指文字や, 手の回転・移動を必要とするものを判別対象に加え, 伴って悪化することが予想される, 検出精度の低下や, 検出時間の増加などを抑える手法を前項のアルゴリズムの選定と合わせて行っていく.

- GROVEデバイスの両手化

  GROVEデバイスは主に片手のみでの運用を前提としているが, これを両手化することで, 手話の検出等を行うことを検討する.

- 身体能力拡張性の追求

  最近, 例えばMITの 5 指の補助をするロボット指の追加の研究[6]など, ロボットを用いて身体を拡張する研究である「Augmented Human」の研究に注目が集まっている. GROVEデバイスを用いて, ロボットハンドを精密に操作し, あらゆる動作の補助として用いることで, 身体能力を拡張する研究を行っていきたい.

- サーバー等との連携

  今後, 複数のGROVEデバイスを同時に用いた時, デバイスの動作等を共通のサーバー上に送信し, データを記録し, 解析することで検出精度等の向上を図りたい. また, ユーザー自身のライフログ（生活の記録）の管理等にも用いることのできるアプリケーションの開発を併せて行う.

- UIの向上

  現在のスマートフォン側アプリケーションのUIはシンプルさに主眼を置いており, 指の曲げ具合を色で表現しているが, 今後は誰にでも直観的に理解できる3Dモデル等で表示することで, より分かり易いUIを構築したい.

- IME化

  現在のアプリケーションでは, 専用のアプリケーションを起動した上でのジェスチャー

の検出・伴ったアクションを行っているが, 今後は, より親側のレイヤー上で動作するIMEアプリケーションを開発し, 専用ではないあらゆるアプリケーションの操作に使役できるようにする.

- 握手によるデータの送受信

GROVEデバイスの手の平側にNFC通信モジュール等を設置することで, デバイスを装着したもの同士が握手することで, 図5.1の様に, メールアドレスの交換や名刺交換などといった, データのやり取りを行うことが出来るように拡張する予定である.

図 5.1 GROVEを用いたアドレス交換

本論文は, 筆者が神戸市立工業高等専門学校 電子工学科, 並びに, 専攻科 電気電子工学専攻において, 若林研究室において行った研究をまとめたものである.

本研究に関して終始ご指導ご鞭撻を頂きました本学若林茂教授に心より感謝致します.

また研究に際し, ロボットハンドを製作するための3Dプリンタを貸借していただいた本学笠井正三郎教授に感謝致します.

## 参考文献


[1] 日本経済新聞 : “小学生がホームから転落,あわや事故,JR四ッ谷駅”, “http://www.nikkei.com/article/DGXNASDG27056_X20C13A5CC1000/”, 2015年1月25日閲覧

[2] 独立行政法人　国立特別支援教育総合研究所 : “手話の指導”, “http://www.nise.go.jp/cms/13,884,44,175.html”,2015年1月25日閲覧

[3] Pranav Mistry,Pattie Maes: “SixthSense: a wearable gestural interface”, ACM SIGGRAPH ASIA 2009 Sketches Article No. 11(SIGGRAPH ASIA 2009)

[4] Chris Harrison,Hrvoje Benko,Andrew D.Wilson: “OmniTouch:Wearable Multitouch Interaction Everywhere”, UIST ’11 Proceedings of the 24th annual ACM symposium on User interface software and technology, pp.441-450(2011)

[5] 鷺坂隆志,大村吉幸,長久保晶彦,國吉康夫,尾崎和行 :
「皮膚の変形に追従する高密度触覚センサグローブの開発」,一般社団法人 日本ロボット学会,日本ロボット学会誌 30(7), pp.711-717(2012-09-15)

[6] Faye Y. Wu,Harry H. Asada :
「Bio-Artificial Synergies for Grasp Posture Control of Supernumerary Robotic Fingers 」,Massachusetts Institute of Technology ,Robotics: Science and Systems X,pp.27,2014

# 付 録A　付録

以下に本研究において参考にした指文字一覧の表を添付する.

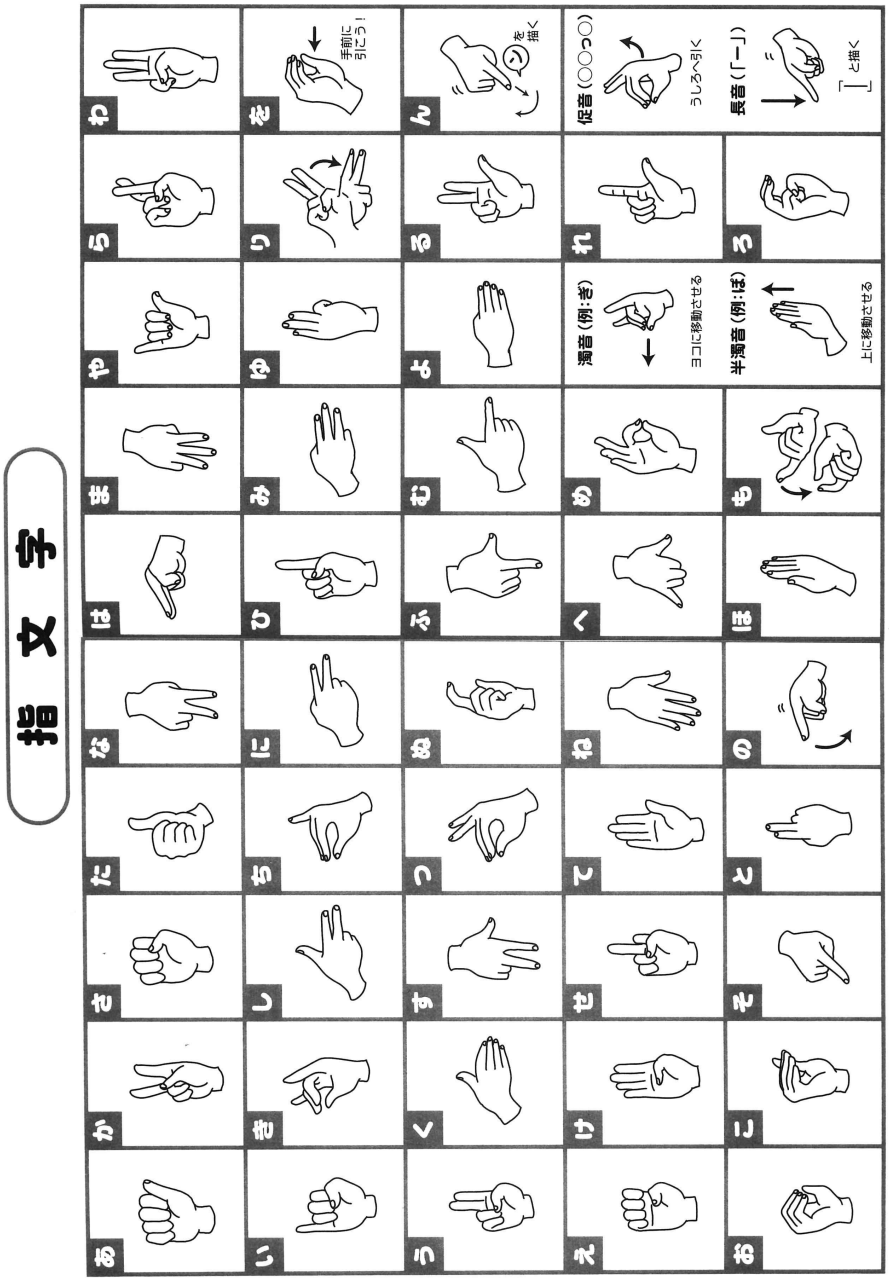


図 A.1 指文字一覧

# 付録2

以下に, 測定精度の検討で検出した, GROVEの測定精度について検証した際にまとめた表を添付する.

なお, 表内のデータの単位はすべて秒である.

表 B.1 GROVEシステムの検出精度(しきい値70, 1回目)

| | 1回目 | 2回目 | 3回目 | 4回目 | 5回目 |
|---|---|---|---|---|---|
| あ | 1.21 | 1.94 | 1.96 | 1.27 | 0.98 |
| い | 0.69 | 0.67 | 3.95 | 1.85 | 4.16 |
| う | 1.28 | 1.03 | 0.30 | 2.14 | 0.84 |
| え | 1.50 | ×(お) | 1.15 | 0.81 | 2.34 |
| お | 0.60 | 0.31 | 2.50 | ×(え) | 1.20 |

表 B.2 GROVEシステムの検出精度(しきい値70, 2回目)

| | 1回目 | 2回目 | 3回目 | 4回目 | 5回目 |
|---|---|---|---|---|---|
| あ | 0.67 | 0.57 | 9.27 | 2.04 | 1.52 |
| い | 1.39 | 1.34 | 6.09 | 3.13 | 7.17 |
| う | 1.00 | 1.17 | 0.56 | 3.30 | 1.40 |
| え | 5.94 | 2.25 | 1.99 | 1.51 | 4.64 |
| お | ×(え) | 2.55 | 1.10 | ×(え) | ×(え) |

表 B.3 GROVEシステムの検出精度(しきい値100, 1回目)

| | 1回目 | 2回目 | 3回目 | 4回目 | 5回目 |
|---|---|---|---|---|---|
| あ | 0.92 | 2.54 | 1.75 | 2.79 | 1.03 |
| い | 0.16 | 0.74 | 0.56 | 0.49 | 0.35 |
| う | 0.92 | 1.93 | 2.91 | 0.29 | 0.18 |
| え | 0.86 | 1.32 | 0.63 | ×(お) | 0.88 |
| お | ×(え) | 0.92 | ×(え) | 1.15 | 1.72 |

表 B.4 GROVEシステムの検出精度(しきい値100, 2回目)

| | 1回目 | 2回目 | 3回目 | 4回目 | 5回目 |
|---|---|---|---|---|---|
| あ | 1.03 | 1.09 | 1.21 | 1.56 | 1.57 |
| い | 0.57 | 1.64 | 1.38 | 0.63 | 0.29 |
| う | 0.77 | 1.33 | 0.37 | 1.48 | 0.74 |
| え | 1.89 | 1.21 | 2.14 | 1.26 | 0.73 |
| お | ×(え) | 2.29 | ×(え) | 1.06 | ×(え) |