

筑波大学大学院博士課程

システム情報工学研究科修士論文

触り方を検出可能な人体通信システムの開発

近藤 勇斗

修士（工学）

（コンピュータサイエンス専攻）

指導教員 高橋 伸

2016年3月

概要

CPU やメモリ、通信モジュールの小型化・高性能化に伴い、ユビキタスコンピューティング環境が実現されつつある。身の回りに存在する物でもセンサノードを取り付けることで、計算能力や通信能力を持つスマートオブジェクトになる。ユビキタスインタラクションではユーザは使い方を意識することなく、直感的な動作によって通信を行えることが望ましい。物体に触るという行為は人間にとって自然な動作であり、直感的であるといえる。そこで、本研究では身の回りのスマートオブジェクトと簡潔かつ直感的に通信できることを目的とした、触り方を検出可能な人体通信システムを開発した。端末間で通信を行うためには端末間の接続設定が必要である。例えば、同じネットワークに接続するという設定やどの端末と接続するか設定を行う必要がある。これに対し、人体通信では人体に触れている端末同士が人体を通じて直接信号のやりとりを行うことができる。そのため、触れるという行為がそのまま接続設定の代わりとなり、予め接続設定を行う必要がなくなる。また、触り方を検出することで触り方に応じてシステムの動作を変えることができ、触り方に応じたデータを通信することができる。

目次

第1章	はじめに	1
1.1	背景	1
1.2	目的	1
1.3	アプローチ	2
1.4	構成	2
第2章	関連研究	3
2.1	人体通信についての研究	3
2.2	タッチ認識についての研究	4
2.3	複数端末間連携についての研究	5
第3章	触り方を検出可能な人体通信システム	6
3.1	システム構成	6
3.2	ウェアラブルデバイス	7
3.3	スマートオブジェクト	7
3.4	触り方の検出と人体通信の連携	7
3.5	ユースケース	7
3.5.1	カップに入ったドリンクの温度表示とライフログの記録	7
3.5.2	棚の中身の閲覧と開錠	8
3.5.3	IDと触り方の組み合わせによるパスワード強化	8
第4章	実装	9
4.1	ウェアラブルデバイス	9
4.2	スマートオブジェクト	9
4.3	人体通信	12
4.3.1	人体通信に利用する信号の選定	12
4.3.2	通信回路	17
4.3.3	通信プロトコル	17
4.4	触り方のセンシング	17
4.4.1	周波数掃引	17
4.4.2	センシング回路	19
4.5	触り方の登録と認識	22

4.6	回路の切り替え	25
4.7	システム処理の流れ	25
4.8	アプリケーション：触り方に応じた画像通信	28
4.8.1	スマートオブジェクトのアプリケーション	28
4.8.2	ウェアラブルデバイスのアプリケーション	28
4.8.3	データ通信の仕組み	28
第5章	評価	33
5.1	実験 1:素材の違いによるセンシングデータの測定	33
5.1.1	被験者	33
5.1.2	実験内容	33
5.1.3	実験結果	36
5.2	実験 2:触り方検出の認識率	38
5.2.1	被験者	38
5.2.2	実験内容	38
5.2.3	実験手順	38
5.2.4	実験結果	40
5.3	実験 3:システム利用	51
5.3.1	被験者	51
5.3.2	実験内容	51
5.3.3	実験手順	51
5.3.4	実験結果	51
第6章	結論	53
	謝辞	54
	参考文献	55

目次

3.1	システム構成図	6
4.1	ウェアラブルデバイス	10
4.2	スマートボックス	11
4.3	Arduino をノート PC で駆動させた場合の人体通過前後の各周波数の信号波形 (1/2)	13
4.4	Arduino をノート PC で駆動させた場合の人体通過前後の各周波数の信号波形 (2/2)	14
4.5	Arduino をスマートフォンで駆動させた場合の人体通過前後の各周波数の信号波形 (1/2)	15
4.6	Arduino をスマートフォンで駆動させた場合の人体通過前後の各周波数の信号波形 (2/2)	16
4.7	通信回路の回路図	18
4.8	回路素子の違いによるセンシングデータの比較	20
4.9	センシング回路の回路図	21
4.10	触り方の登録及び識別を行うアプリケーション	23
4.11	触り方の登録及び識別を行うアプリケーション (Touch を認識した場合)	23
4.12	軸を変換したアプリケーション	24
4.13	軸を変化したアプリケーション (Touch を認識した場合)	24
4.14	システム処理の流れ	27
4.15	スマートオブジェクトのアプリケーション	30
4.16	ウェアラブルデバイスのアプリケーション (1/2)	31
4.17	ウェアラブルデバイスのアプリケーション (2/2)	32
5.1	実験で使用した金属板	34
5.2	実験環境	34
5.3	各触り方 (銅板)	35
5.4	各素材毎の触り方の違いによるセンシングデータの比較	36
5.5	各触り方毎の素材の違いによるセンシングデータの比較	37
5.6	各触り方 (スマートボックス)	39

表目次

4.1	回路切り替えに用いるパケット	25
5.1	各物質の抵抗率	33
5.2	銅板に対する触り方の識別結果(被験者 A)	41
5.3	銅板に対する触り方の識別結果(被験者 B)	42
5.4	銅板に対する触り方の識別結果(被験者 C)	43
5.5	銅板に対する触り方の識別結果(被験者 D)	44
5.6	銅板に対する触り方の識別結果(全体の平均)	45
5.7	スマートボックスに対する触り方の識別結果(被験者 A)	46
5.8	スマートボックスに対する触り方の識別結果(被験者 B)	47
5.9	スマートボックスに対する触り方の識別結果(被験者 C)	48
5.10	スマートボックスに対する触り方の識別結果(被験者 D)	49
5.11	スマートボックスに対する触り方の識別結果(全体の平均)	50
5.12	タスク成功率	52

第1章 はじめに

1.1 背景

CPUやメモリ、通信モジュールの小型化・高性能化に伴い、ユビキタスコンピューティング環境が実現されつつある。身の回りに存在する物でもセンサノードを取り付けることで、計算能力や通信能力を持つスマートオブジェクトになる。また、インタラクション技術の進歩はGUIのような伝統的な手法から日常的な物体を連携させて扱うようなユビキタスインタラクションへと移り変わってきた [1]。ユビキタスインタラクションで求められる重要な点は、物体やそこに取り付けられた外部のコンピューティング基盤とお互いに通信ができる点である [2]。ユーザは使い方を意識することなく、直感的に動作を行うことで通信を行うことができることが望ましい。

我々は、ユーザが使い方を意識することなく直感的に動作を行うことで通信ができることを目指し、複数端末間において簡潔で分かりやすい情報移動を行うための手法を提案した [3][4]。この手法はタッチパネルとの人体通信を利用し、ユーザが身につけたウェアラブルデバイスとタッチパネル端末間で画像などの情報を移動させるものである。この手法は人体を通信経路として使用する。この手法を利用しスマートオブジェクトと通信を行うことができるようにすることで、より使いやすいインタラクションを実現できると考えた。スマートオブジェクトへの触り方を検出できるようにし、その触り方に応じて通信するデータを変えることで、ユーザは直感的な動作のみで簡潔に通信を行うことができると考えられる。

1.2 目的

本研究ではスマートオブジェクトへの触り方を検出できるようにし、その触り方に応じて通信するデータを変えることで、ユーザが直感的な動作のみで簡潔に通信を行うことのできるシステムを開発することを目的とする。通信を行うためにユーザによる煩雑な設定を不要にし、物体に触るという日常の中で行う直感的な動作により通信を行う。なおかつ、触り方に応じて通信するデータを変えることで、ユーザの意図に沿ったインタラクションが行えるようになる。

1.3 アプローチ

前節の目的を達成するため、本研究では交流信号の周波数掃引による触り方の検出及び、それに対応したデータを人体通信により伝達する方法を行う。交流信号の周波数掃引を行うとユーザの触り方によって信号の反応が変化する。これを利用することでユーザの触り方を検出することが可能になる。人体通信とは人体をケーブルのように利用し、人体に電気信号を流すことで行う通信手法のことである。人体に触れている端末同士で人体を通じて直接信号のやりとりを行うため、予めどの端末を接続するかという設定が不要になる。これらのアプローチにより触り方を検出可能な人体通信システムを開発する。

1.4 構成

本論文の構成は以下の通りである。本章では、本研究における背景を述べ、それを踏まえて本研究の目的とアプローチを述べた。第2章では、本研究に関連する研究やサービスを紹介し、本研究との比較によって本研究の位置付けを行う。第3章では触り方が検出可能な人体通信システムの概要と構成を述べ、続く第4章でシステムの実装を説明する。第5章では実験を通じて行ったシステムの評価・性能について述べ、第6章で結論を述べる。

第2章 関連研究

2.1 人体通信についての研究

人体を伝送路として利用する人体通信は、通信相手に触れることで伝送路を確立し通信を実現するため、ユビキタスネットワーク社会における新たなマンマシンインタフェースとして考えることができる。人体通信は、1996年 Zimmerman によりウェアラブル・コンピューティングにおける CPU と各種デバイス間の通信に利用することが提案された [5]。これを契機として、人体通信に適するアプリケーションと実現方式の検討が開始された [6][7]。人体を伝送路として利用するためにその特性についても研究が行われてきた [8][9][10]。人体通信の信号伝送方式としては主に2つの方式が提案されている。

1. 電界方式

電界方式は次のような方式である。「電界方式送信機により人体表面に電界を誘起させ、その電界に信号をのせ、受信機で電界を検知し受信する。これは誘電体である人体を利用し、電界を人体近傍に発生させることで人体に電界がまとっているようになる。この方式であれば絶縁体を通した状態でも信号伝送が可能になる。よって、利用状況・方法が多く、送信・受信機の設置位置の制限が減ることになる。しかし、人体に電界を発生させるとなると周りの環境・人体の状態による雑音の影響が大きい。さらに信号を検知することが難しく、人体に誘起した電界の大半が地面に流れてしまい受信機で受ける電界が微弱なものとなる」[8]。このように、人体が触れなくても通信を行うことができるが、意図せず通信を行ってしまう可能性が生じるというデメリットもある。

2. 電流方式

電流方式は次のような方式である。「電流方式はその名の通り、人体に微弱な電流を流すという方法である。通信としての原理は有線通信と同様であり、通信回路も比較的簡単なものができる。この方式は電界方式よりも比較的環境・人体の状態による雑音にも強い。しかし、電流を用いたとなると送信端末からの電極は伝送路とする人体に必ず接触させておく必要があり、絶縁体を挟んだ通信が不可能である。そのため利用方法・設置位置の制限がかかってくる」[8]。このように通信の際には必ず電極に触れる必要があるが、電流は人体の外部へは漏れないため、情報漏洩しにくいというメリットもある。

本研究ではスマートオブジェクトに触れるという動作によって通信を行うため、電流方式を採用することとする。

人体通信では通信の安定のために閉ループの構成が重要である。ある機器 A・B 間を有線接続にて通信させる場合、信号線と基準線の少なくとも 2 本の配線が必要になる。この 2 本の配線により機器 A・B 間に閉ループを構成する。これにより電流が流れ通信を行うことができる。人体通信は、ケーブルの代わりに機器 A・B 間の人体を信号線として、大地アースと静電結合することで閉ループ回路を構成する。これにより、電気の流れが発生し通信を行うことができる [11]。

人体通信を利用したインタフェースとして Wearble Key[12]がある。これは Wearable Key を身につけたユーザが PC 等の端末に触れることで人体を通じて ID を送り、電子ロックを解除するといったものである。同様に吉田らは人体通信を利用してアフォーダンスの良いドアエントリーシステムを開発した [11]。Enhanced Touch[13]は手首にデバイスを装着して、人体通信により人と人との接触を検出する。Diamondtouch[14]はテーブルトップディスプレイにおいて人体通信を利用して、タッチ点を検出するシステムである。マルチユーザが利用する場合にはどのユーザがディスプレイに触れているかを識別することができる。また、CapNFC[2]は静電容量のセンシングや人体通信を利用し、日用品と近距離でのユビキタスインタラクションを可能にする。

本研究はこれらの研究と類似する点が多い。人体通信を利用し、身の回りのものとユビキタスインタラクションを可能にすることを目指す。本研究が先行研究と異なる点は身の回りのものにただ触れるだけでなく、触り方を検出できるようにし、触り方の違いによってシステムの動作を変える点である。触るという直感的な行為のみで通信をするデータを選択するなど、システムの動作の種類を増やすことができる。

2.2 タッチ認識についての研究

タッチ認識のためのタッチセンサには静電容量方式 [15][16]、感圧方式 [17]、光学方式 [18]、[19]、音響方式 [20] 等、様々な形態が提案されている。タッチセンサはセンサに指が触れているかどうかを認識するものでありスイッチと同じような役割である。しかし、複数の微小センサをアレイ状に配置しやすいという点で機械式スイッチとは異なる。SmartSkin[15]やDiamondTouch[14]はセンサをアレイ状に配置することでタッチ位置やジェスチャを検出できるようにし、入力自由度を高めたものである。このように多数のタッチセンサを用いることでタッチ位置やジェスチャを検出できるが多数のタッチセンサを用いるには多数の複雑な結線が必要であり、回路規模や消費電力の増加を招く。そこで、センサの数を減らしても高い入力自由度を持つことが重要となる。

Touché[16]は、タッチの状態によってインピーダンスの周波数特性が異なることを利用し、1つの電極を用いるだけで把持状態の識別やどのユーザの指が接触しているのかの認識を行うことができる。交流信号は人体を通過し、その信号量は信号周波数に強く依存する。これは、周波数が異なると交流信号が体内の異なる経路を流れるからである。交流信号は最小のインピーダンスの経路を流れる。人体は複雑な構造をしており、例えば、筋肉、脂肪、骨には異なる抵抗と容量特性がある。交流信号の周波数が変わると、幾つかの組織は電荷の流れ

を妨げるようになり、一方で幾つかの組織は電荷の妨げが小さくなる。従って、信号の流れは周波数によって変化する。そのため、周波数掃引を行うことにより、把持状態の識別やどのユーザの指が接触しているのかの認識を行うことができる [16][21]。同様な認識は振動を用いても可能である [20]。

本研究はこうした入力の自由度が高いタッチ認識技術を利用してユーザの触り方を検出する。加えて、検出した結果に応じてシステムの動作を変え、人体通信によりデータを伝達するシステムを開発する。

2.3 複数端末間連携についての研究

複数端末間の連携として Hinckley[22] や Seifert[23] は端末の外周同士を接触させることにより端末同士の連携を行い、接触を行っている辺において画面を繋げることでユーザは複数の画面が一体化したかのようにインタラクションを行うことができる手法を提案した。Pick-and-Drop[24] はディスプレイ上のオブジェクトを ID が割り当てられたペンデバイスで「つまみ上げ」、他のコンピュータに「置く」ことで情報の移動を実現している。また、Memory stones[25] ではマルチタッチ対応のタッチパネルに触れた複数指の位置を検出し、これが形作る多角形の形状から端末のペアリングを行うことで、複数コンピュータ間での情報のやりとりを実現する。

本研究では上記の先行研究と同様に、直感的な操作で端末の接続と情報転送の両方がシームレスに行える手法を実現する。異なる点は、先行研究では情報移動の手法として無線通信を利用したのに対し、本研究では人体通信を用いる点である。無線通信では予めコンピュータを無線ネットワークに接続する、あるいはどの端末と接続するかという設定が必要になる。しかし、人体通信を用いることでユーザが触れることで端末間の接続経路が確立される。そのため、予め無線ネットワークに接続したりどの端末同士を接続するかという設定を行う必要がなくなる。

また、人体通信を利用することで端末自体を近づける必要がなくなる点も利点となる。例えば、メガネ型のウェアラブルデバイスやベルト型のウェアラブルデバイスを別の端末に接触させる、あるいは近づけるというのは物理的に困難な場合が考えられる。一方、人体通信では端末が人体に接触する必要という制約があるものの、端末自体を近づける必要はなく、ユーザが触れている端末同士が接続される。そのため、端末同士を近づけることが困難な場合でも簡潔に端末の接続及び通信が可能となる。

第3章 触り方を検出可能な人体通信システム

本研究では身の回りに存在するスマートオブジェクトと簡潔で分かりやすい手法で通信を行うために、触り方を検出可能な人体通信システムを開発する。ウェアラブルデバイスを身につけたユーザがスマートオブジェクトに触れた時、スマートオブジェクトの機能によりユーザがどんな触り方をしたかを検出する。検出した触り方に応じてシステムの動作や通信するデータを変える。データの通信は人体通信で行う。

3.1 システム構成

本システムの構成を図 3.1 に示す。本システムはウェアラブルデバイスとスマートオブジェクトから構成される。本研究で開発するウェアラブルデバイスはコンピュータと通信モジュールを搭載したユーザが身につけて利用するデバイスであり、スマートオブジェクトは物体にコンピュータ、通信モジュール、センシングモジュールを搭載させたものである。本研究では導体の物体をターゲットとする。ウェアラブルデバイスとスマートオブジェクトが搭載する通信モジュールはどちらも同じものであり、通信モジュールの機能により人体通信を実現する。スマートオブジェクトに搭載したセンシングモジュールはユーザの触り方を検出する機能を持ち、検出した触り方に応じてシステムの動作や通信するデータを変える。

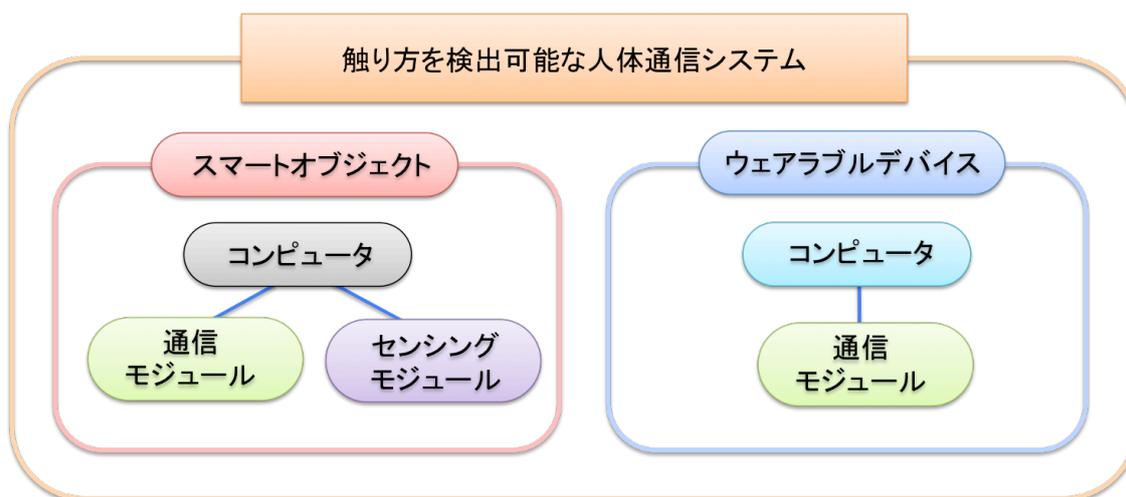


図 3.1: システム構成図

3.2 ウェアラブルデバイス

ウェアラブルデバイスはコンピュータと通信モジュールを搭載したユーザが身につけて利用するデバイスである。ウェアラブルデバイスには電極を取り付け、電極がユーザの体に接触するように装着する。通信モジュールの機能によりこの電極を通じてユーザの体に信号を流したり、ユーザの体を伝わって送られてきた信号を受け取ったりすることで人体通信を実現する。

3.3 スマートオブジェクト

スマートオブジェクトはコンピュータ、通信モジュール、センシングモジュールを搭載した導体の物体である。文房具や家具、ドア等にコンピュータ、通信モジュール、センシングモジュールを搭載することでスマートオブジェクトとすることを想定する。物体に搭載したセンシングモジュールによりユーザの触り方を検出する。その検出結果に応じて、通信モジュールの機能によりユーザが触った部分から人体通信によりデータを伝送する。

3.4 触り方の検出と人体通信の連携

ユビキタスインタラクションではユーザは使い方を意識することなく、直感的に動作を行うことで通信を行えることが望ましい。物体に触るという行為は人間にとってごく自然な動作であり、直感的であると言える。従って、システム動作のアクションとして触るという行為を用いることはユーザにとって使いやすいシステムを実現することにつながると考えられる。

ここで重要な点としては通信を行う場合には端末間の接続設定が必要な点である。例えば、Wi-Fiを利用する場合には同じネットワークに接続する設定が必要である。Bluetoothではどの端末と接続するかを設定を行う必要がある。これに対し、人体通信では人体に触れている端末同士が人体を通じて直接信号のやりとりを行うことができる。そのため、触れるという行為がそのまま接続設定の代わりとなるため、予め接続設定を行う必要がなくなる。そのため、触り方の検出と人体通信を組み合わせることで、触り方に応じて動作・送るデータを変えらるというように直感的な動作によって様々なインタラクションを行うことができ、かつその行為のみで通信を行うことができる。従って、触り方を検出可能な人体通信システムはユーザにとって簡潔で分かりやすいシステムであると考えられる。

3.5 ユースケース

3.5.1 カップに入ったドリンクの温度表示とライフログの記録

搭載されたセンサにより、ドリンクを注ぐとそのドリンクの種類やカロリー、タンパク質、温度等のデータを取得できるカップがあるとすると、ユーザはドリンクを飲む前にカップの取っ

手に1本の指で触れる。するとドリンクの温度がウェアラブルデバイスへ送られ表示される。これにより、ユーザはドリンクを飲む前に適温であるかを知ることができる。取っ手の部分であればドリンクの温度に影響されず、触っても安全である。また、情報はウェアラブルデバイスで表示するためカップにディスプレイ等の表示機能がなくても使用可能というメリットもある。

温度を確認した後は、ドリンクを飲むためにカップの取っ手を握る。すると、カップに入ったドリンクの種類やカロリー、タンパク質等の情報が身につけたウェアラブルデバイスに送信され、ライフログとして記録される。カップ自体にライフログを記録する訳ではないため、専用のカップを用意せずとも、センシングしたデータを送るだけで個人ごとのライフログを記録することができる。また、センシングしたデータの中には温度等ライフログには特に必要のないデータも存在すると考えられる。触り方に応じて表示するだけの情報と記録まで行う情報と分けることで必要な情報のみ記録することが可能となる。

3.5.2 棚の中身の閲覧と開錠

多数の品物が収納されている棚が存在するものとする。この棚には電子的に鍵がかけられている。品物が多い為、棚を開けてみても目的の品物を見つけるのは難しい。こうした場合に、棚の取っ手を1本の指で触る。すると、収納されている品物の一覧がウェアラブルデバイスに送られて、目的の品物があるか実際に中を見るよりも分かりやすくなる。目的の品物が棚に収納されていることが確認できたら、今度は取っ手を握る。すると、ウェアラブルデバイスから鍵を開けるためのパスワードが送られ、そのまま簡単に棚を開けることができる。

3.5.3 IDと触り方の組み合わせによるパスワード強化

電子的な鍵がかけられている金庫が存在するものとする。金庫を開けるパスワードの1つであるIDはユーザのウェアラブルデバイスに登録されており、金庫に触れるとそのIDを送信する。しかし、そのIDを送るだけでは金庫は開かない。金庫にはIDとともに特定の触り方が鍵として登録してある。IDだけでなく、触り方も登録してあるものと一致した時、金庫は開く。このように、IDと触り方の組み合わせによりパスワードを強化するといった利用方法も考えられる。加えて、触り方を知っている本人にとっては余計な動作は不要であるため、煩わしさが少ないという利点もある。

第4章 実装

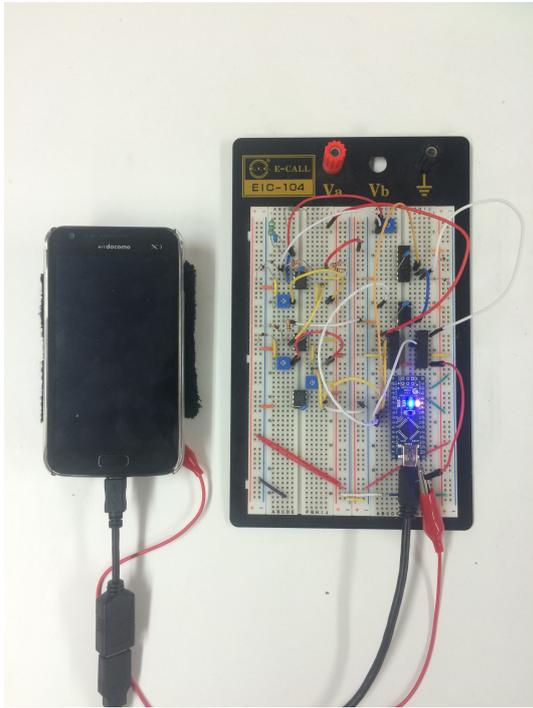
本章では第3章で述べたシステム構成要素について実際に行った実装の詳細について述べる。

4.1 ウェアラブルデバイス

図 4.1 は作成したウェアラブルデバイスである。ユーザが身につけるためにコンピュータとしてスマートフォンを利用した。スマートフォンは OS に Android 4.0.3 を搭載した SAMSUNG 社製の GALAXY S II LTE である。スマートフォンと通信モジュールは USB ケーブルによって接続される。ユーザが身につけられるよう、スマートフォンの裏側にリストバンドを取り付けた。ユーザが使用する際にはこのリストバンドに腕を通して手首に装着するため、リストバンドの内側には電極として銅箔を取り付けた(図 4.1(b))。これにより、銅箔とユーザの体が接触する。この銅箔は通信モジュールの IN/OUT ポートと接続されている。これにより接触部から電気信号の入出力を行うことができるため、人体通信が可能になる。

4.2 スマートオブジェクト

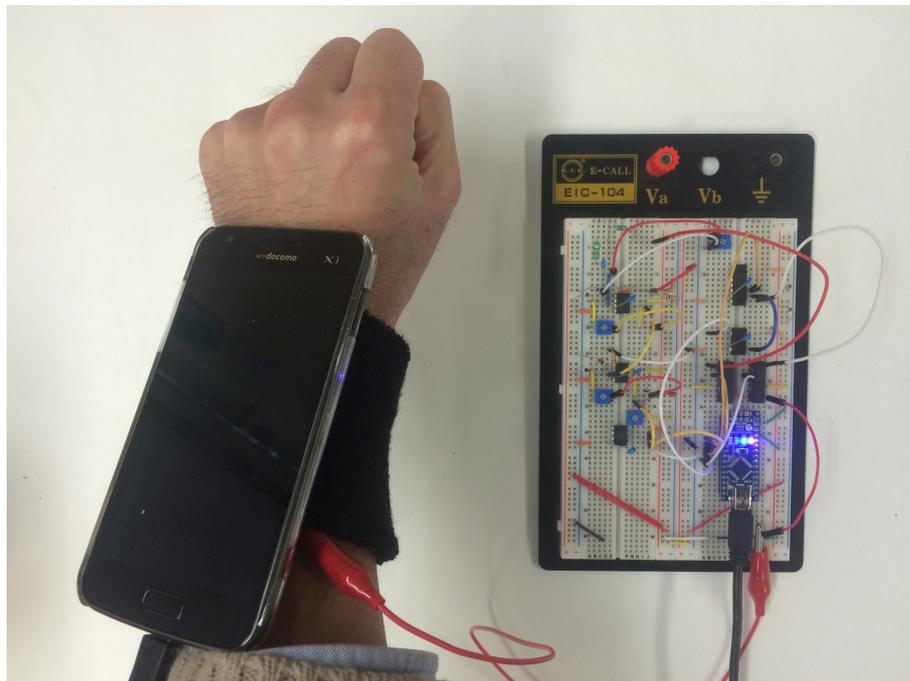
スマートオブジェクトはコンピュータ、通信モジュール、センシングモジュールを搭載した導体の物体である。コンピュータとして Windows7 を搭載した PC を利用した。PC と通信モジュール及びセンシングモジュールは USB ケーブルによって接続される。モジュールの IN/OUT ポートを物体に接続することでスマートオブジェクトとなる。センシングモジュールの機能によってユーザの触り方を検出することができ、なおかつ導体に触れるため、接触部から電気信号の入出力を行うことができ、人体通信が可能になる。図 4.2 はスマートオブジェクトの一例として作成した直方体のプラスチックの周りを銅箔で包んだ導体の物体である。これをスマートボックスと名付ける。図 4.2(b) のようにモジュールの IN/OUT ポートを物体に接続することで、コンピュータ、通信モジュール、センシングモジュールを搭載したスマートオブジェクトとなる。



(a) 表側



(b) 裏側



(c) 装着時

図 4.1: ウェアラブルデバイス



(a) 作成したスマートボックス



(b) 全体像

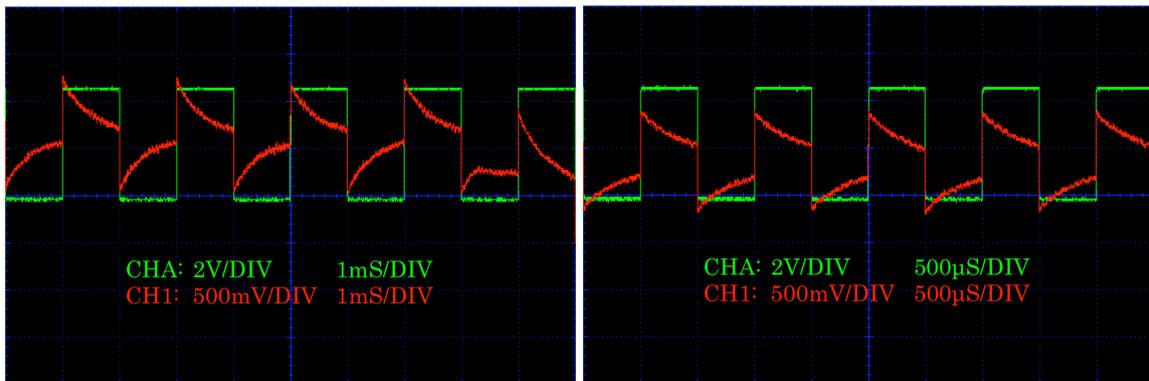
図 4.2: スマートボックス

4.3 人体通信

4.3.1 人体通信に利用する信号の選定

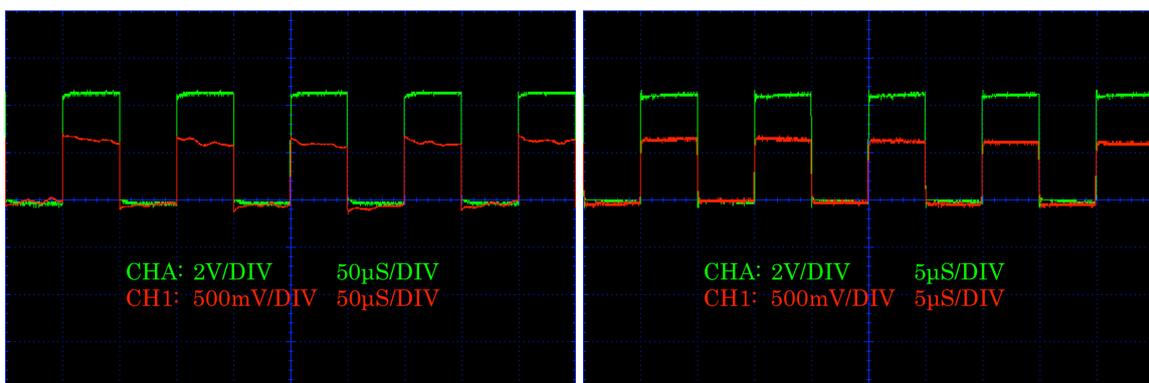
本システムでは人体通信を行うために、信号波を変調して伝送を行う。そこで、どのような周波数の信号が搬送波に適しているかどうかを調べるために基礎実験を行った。実験内容としては内側に銅箔を取り付けたリストバンドを用意し、被験者はこれを右手首に装着する。出力側では Arduino 及びクリスタルオシレータを用いて、電圧約 5V の信号を周波数を変化させて出力する。被験者は信号が出力されるワニ口クリップの金属部分を左手で握る。その出力電圧をオシロスコープで観測する。接続部分を人体のみするために、Arduino を駆動させるために接続したノート PC は電源に接続されていない状態で測定を行った。図 4.3, 図 4.4, は各周波数の場合のオシロスコープで観測した波形である。緑色で示す CHA は入力信号の波形であり、赤色で示す CH1 は出力信号の波形である。実験結果は 100kHz 未満の場合は入力信号に対して出力信号が大きく歪んでいることが分かる。100kHz を超えると入力信号に対して歪みが小さい。一方で、周波数が高くなりすぎると、再び波形が歪むようになるのに加え、出力信号の振幅も小さくなるという結果が得られた。

同様の基礎実験を今度は Arduino を駆動させるためにスマートフォンを使用して行った。図 4.5, 図 4.6 は各周波数の場合のオシロスコープで観測した波形である。ノート PC を使用した場合と同様に緑色で示す CHA は入力信号の波形であり、赤色で示す CH1 は出力信号の波形である。実験結果はこちらも同様の結果となったが、ノート PC で駆動させた場合と比較して、100kHz の場合がやや波形が歪んだ。これらの結果から本研究では搬送波として 1MHz の信号を使用することとした。



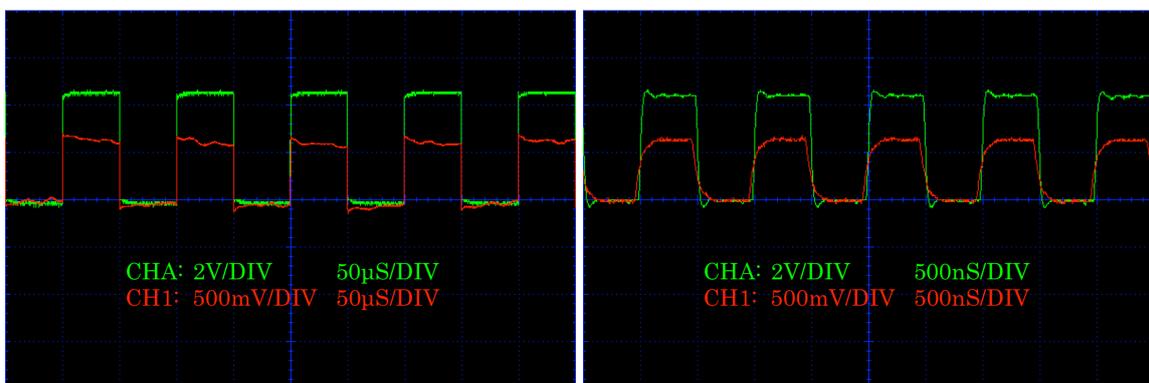
(a) 周波数 500Hz

(b) 周波数 1kHz



(c) 周波数 10kHz

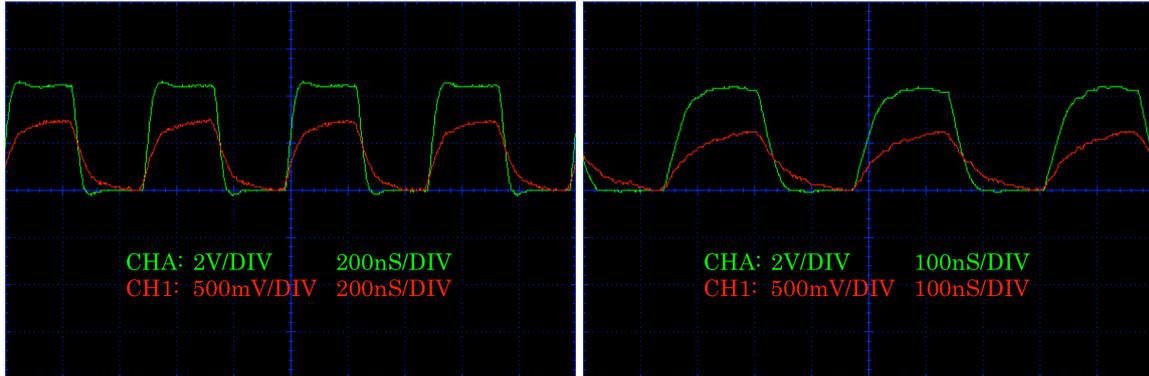
(d) 周波数 100kHz



(e) 周波数 500kHz

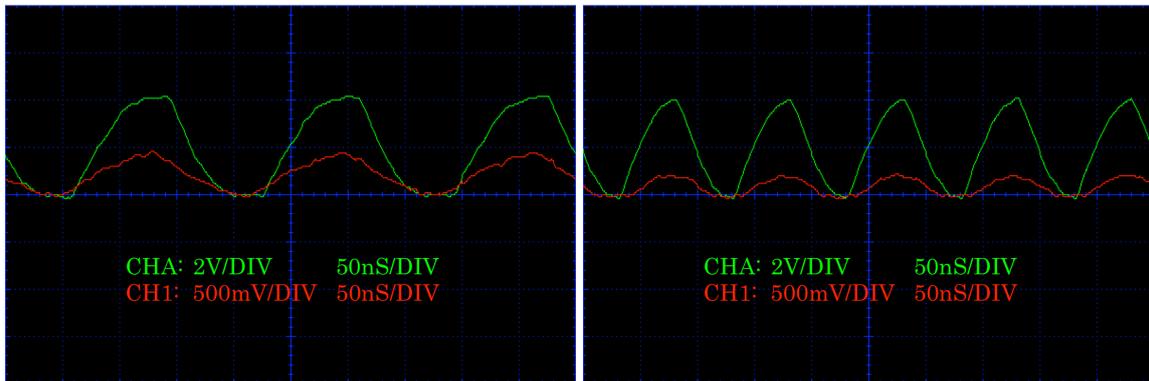
(f) 周波数 1MHz

図 4.3: Arduino をノート PC で駆動させた場合の人体通過前後の各周波数の信号波形 (1/2)



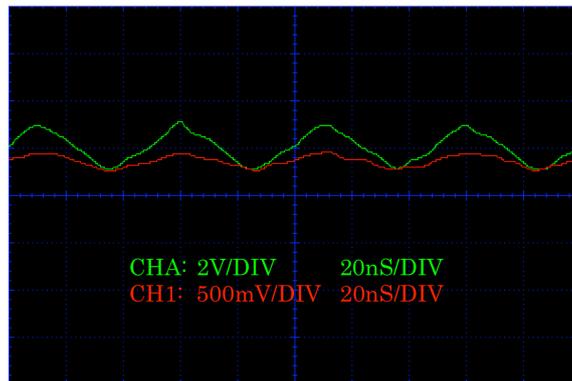
(a) 周波数 2MHz

(b) 周波数 3MHz



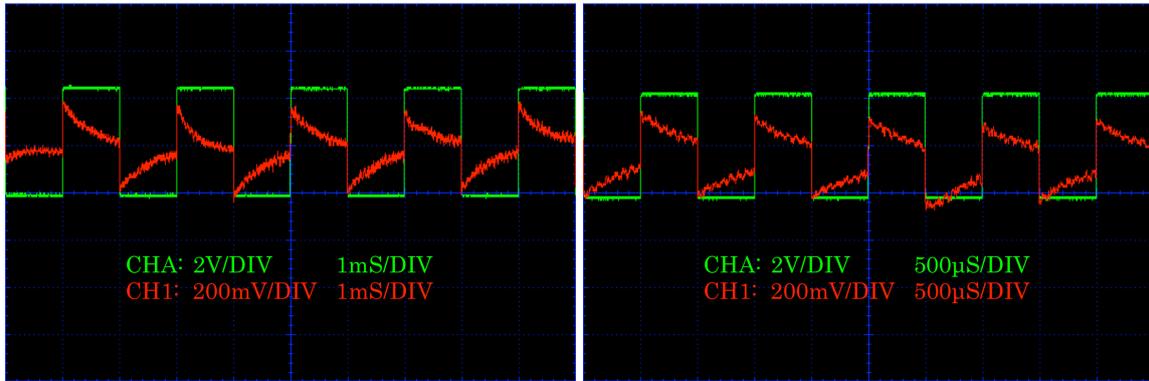
(c) 周波数 6MHz

(d) 周波数 10MHz



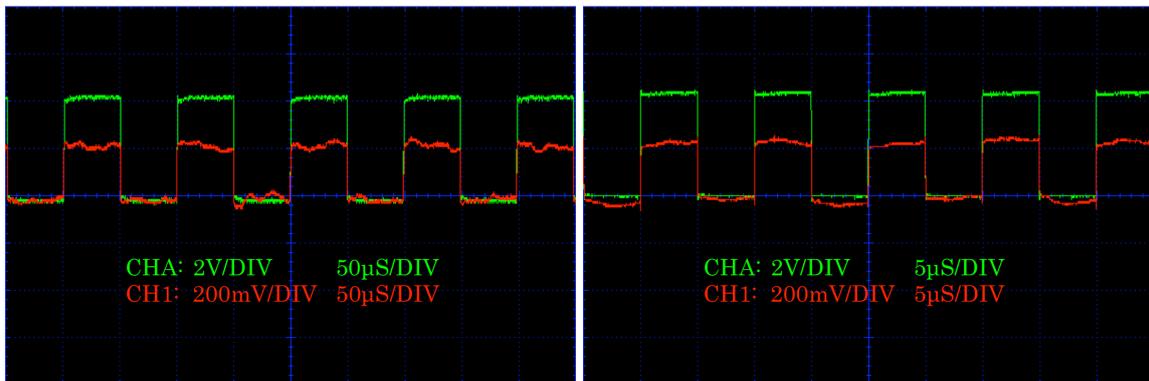
(e) 周波数 20MHz

図 4.4: Arduino をノート PC で駆動させた場合の人体通過前後の各周波数の信号波形 (2/2)



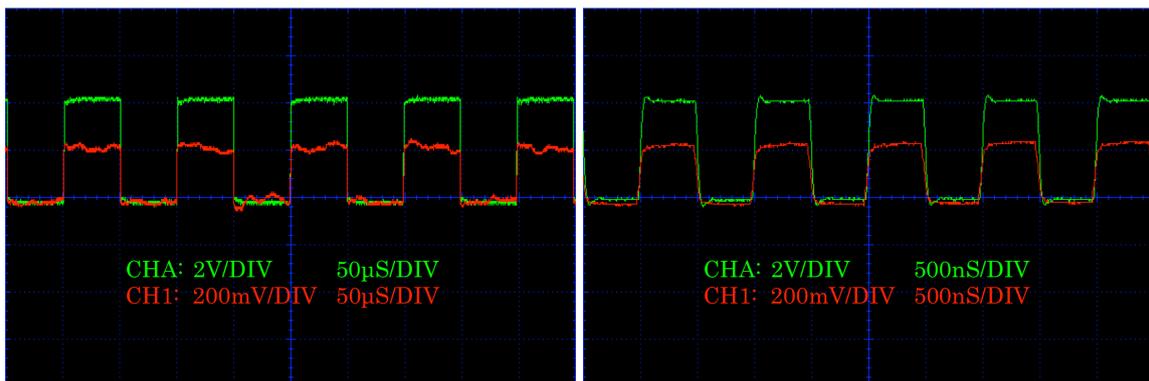
(a) 周波数 500Hz

(b) 周波数 1kHz



(c) 周波数 10kHz

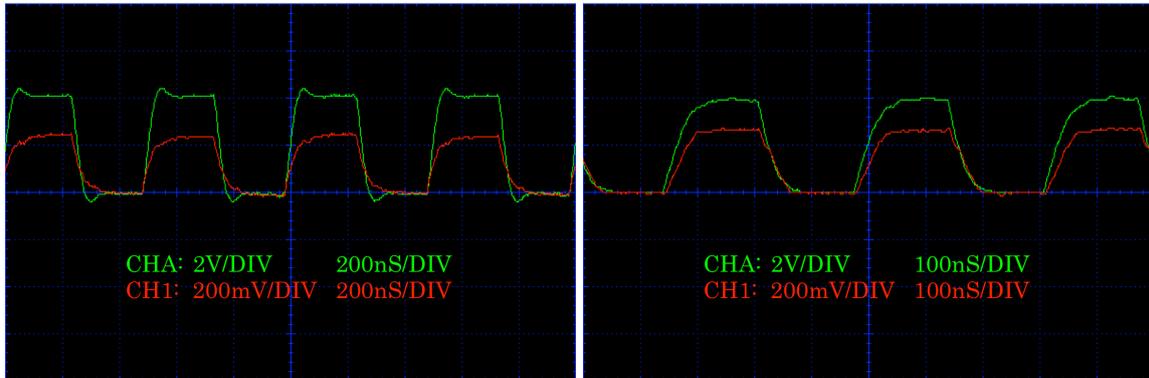
(d) 周波数 100kHz



(e) 周波数 500kHz

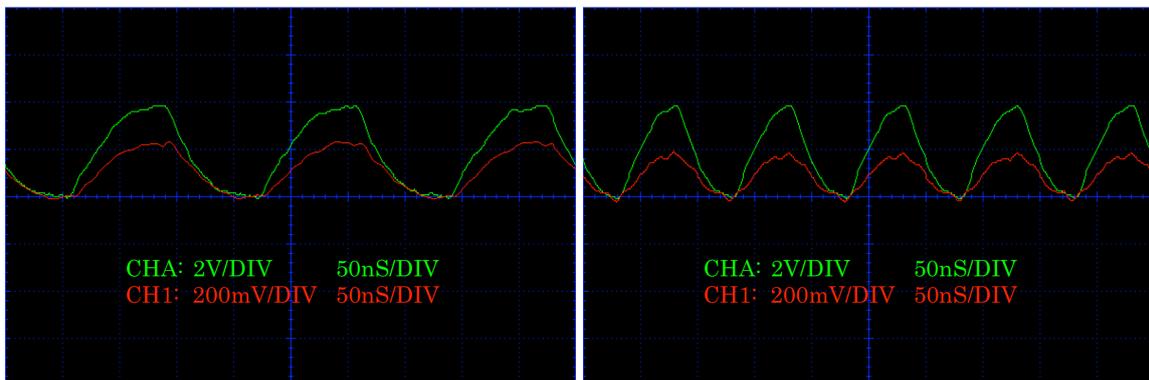
(f) 周波数 1MHz

図 4.5: Arduino をスマートフォンで駆動させた場合の人体通過前後の各周波数の信号波形 (1/2)



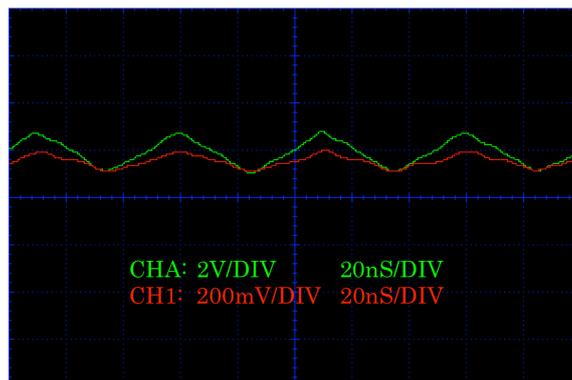
(a) 周波数 2MHz

(b) 周波数 3MHz



(c) 周波数 6MHz

(d) 周波数 10MHz



(e) 周波数 20MHz

図 4.6: Arduino をスマートフォンで駆動させた場合の人体通過前後の各周波数の信号波形 (2/2)

4.3.2 通信回路

双方向通信に対応出来るよう通信回路は送信回路と受信回路を備える。通信回路の制御には Arduino nano 3.1 を使用した。図 4.7 は作成した回路図である。図 4.7(a) は送信回路、図 4.7(b) は受信回路の部分である。送信回路は搬送波を出力するクリスタルオシレータ、搬送波と信号波を合成する AND 回路で構成されている。コンピュータで扱う情報を電圧 5V のデジタル信号へ変換し、Arduino で出力する。出力した信号波をクリスタルオシレータから出力した周波数 1MHz の搬送波により変調する。変調方式は振幅偏移 (ASK) 変調を用いた。受信回路はバンドパスフィルタ、2 段オペアンプ、包絡線検波回路、コンパレータで構成されている。受信した信号を初めに中心周波数 1MHz のバンドパスフィルタにかけ、オペアンプで増幅する。増幅した信号を基準電圧 2.5V のコンパレータにかけデジタル信号へ復調し Arduino で信号を読み取る。送信回路と受信回路を 4.7(c) のように結線し、リレーを用いて送信/受信を切り替えることで半二重通信を行う。これにより人体を通る経路のみで双方向の通信が可能になる。

4.3.3 通信プロトコル

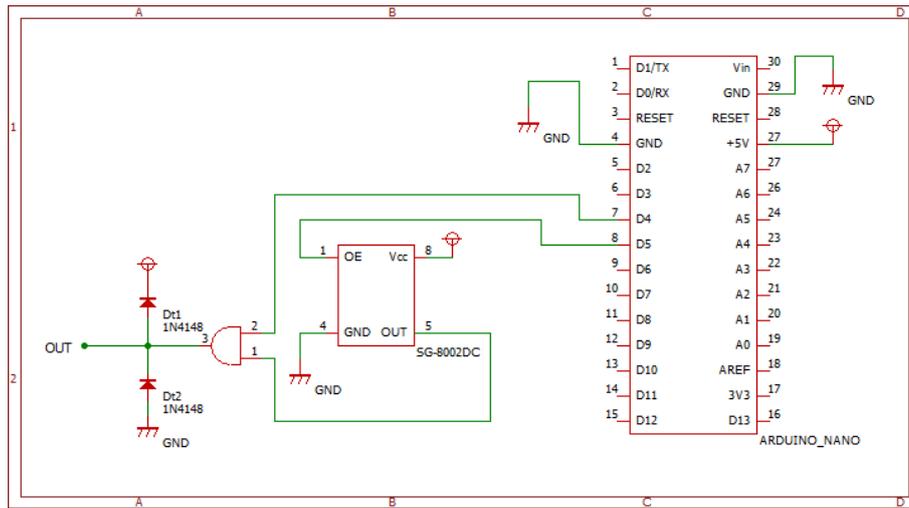
ウェアラブルデバイスとスマートオブジェクトの通信間は全てシリアル通信で行う。通信はベースバンド方式でありボーレートは 57600bps に設定した。Arduino のシリアル通信のボーレートは 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400, 57600, 115200bps がサポートされている。しかし、最大値である 115200bps で通信を行うと、ウェアラブルデバイスとスマートオブジェクト間を人体でなくケーブルで接続した場合でも送信したデータに対して誤ったデータを受信することが頻繁に発生した。一方、57600bps に設定するとウェアラブルデバイスとスマートオブジェクト間をケーブル及び人体で接続した場合でも送信したデータに対して正しいデータを受信することができた。

4.4 触り方のセンシング

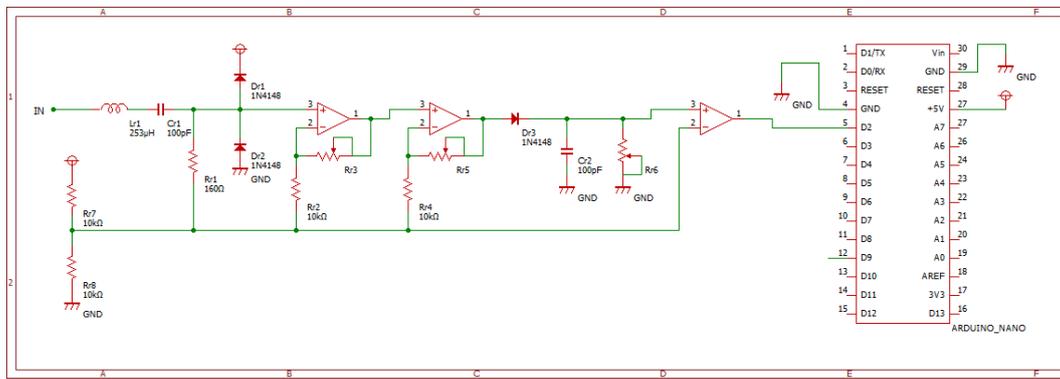
4.4.1 周波数掃引

人体に電気信号を流したとき、信号の流れは周波数によって変化する。そのため、周波数掃引を行うことにより、ユーザがどのように物体に触っているかという情報を得ることが出来る [16][21]。そこで、Touché を Arduino 用に実装した Touche for Arduino: Advanced touch sensing[26] を参考に Arduino で周波数を変化させて信号を出力するプログラムを作成した。Arduino nano 3.1 はクロック周波数 16MHz であり、出力信号の周波数は 16MHz を分周したものになる。従って、出力した周波数は以下の式になる。

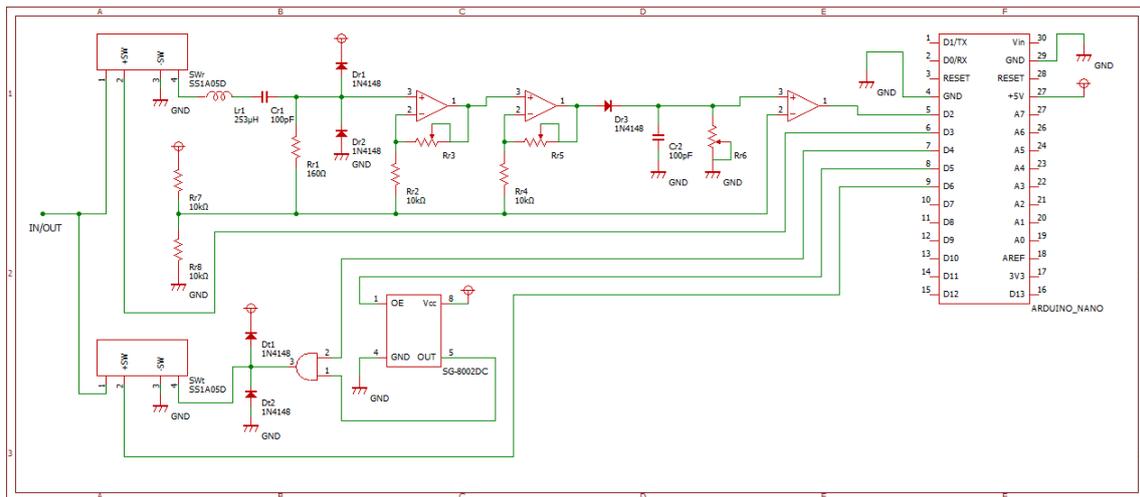
$$f = \frac{16}{x+2} [\text{MHz}] (x \text{ は } 0 \text{ 以上の整数}) \quad (4.1)$$



(a) 送信回路



(b) 受信回路



(c) 通信回路

図 4.7: 通信回路の回路図

本システムでは x を 0 から 126 まで変化させるように設定した。従って、125kHz から 8MHz まで 127 ステップの信号を 1 サイクルとして出力する。この信号は最大値 5V の方形波である。

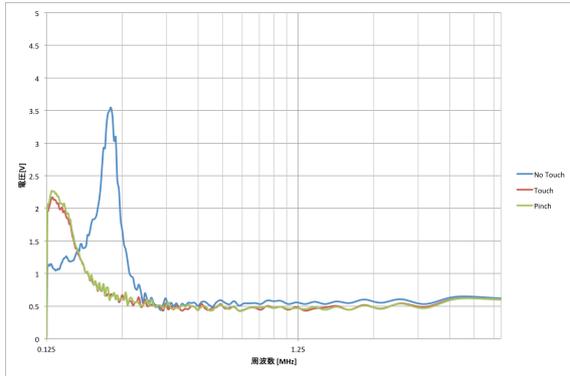
このように出力した信号はセンシング回路を通過し、Arduino の入力ポートへ入力される。各ステップ毎の入力電圧を配列に保存する。1 サイクルの掃引を完了すると Arduino はセンシングしたデータを PC へ送信する。送信するデータは 5V を 1024、0V を 0 として送信する。125kHz から 8MHz までの 1 回の掃引には約 20ms かかる。加えて、Arduino でのセンシングデータを PC へ送信するのに約 162.5ms かかる。従って、1 回のサンプリングには約 182.5ms かかることになり、サンプリングレートは約 5.48Hz となる。

4.4.2 センシング回路

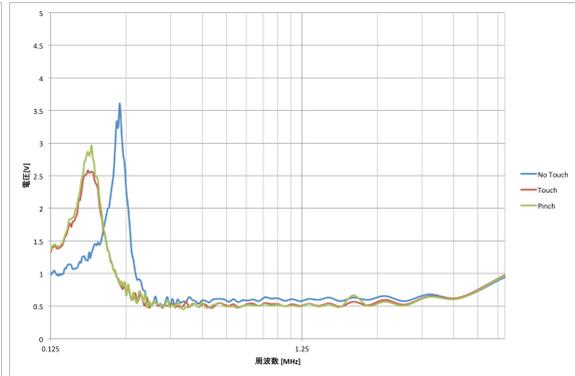
Touche for Arduino: Advanced touch sensing[26] を参考にセンシング回路を作成した。図 4.9(a) はこの回路図である。周波数掃引の際に出力した信号はセンシング回路を通り、Arduino の入力ポートへ入力される。このセンシング回路を通過することで、周波数により入力電圧が変化する。加えて、IN/OUT に接続している物体によってこの波形は変化する。図 4.8(a) は周波数を行った時の波形である。触り方は何も触れていない”No Touch”，1 本指で触れる”Touch”，2 本指で摘むようにして触れる”Pinch”の 3 種類を行った。このようにユーザがスマートオブジェクトに触れることで波形が変化した。しかし、”Touch”と”Pinch”の差はあまり表れなかった。そのため、回路素子を以下のように変更した。

- C2: 10nF → 100pF
- C1: 100pF → 10nF
- R2: 3.3k Ω → 5k Ω

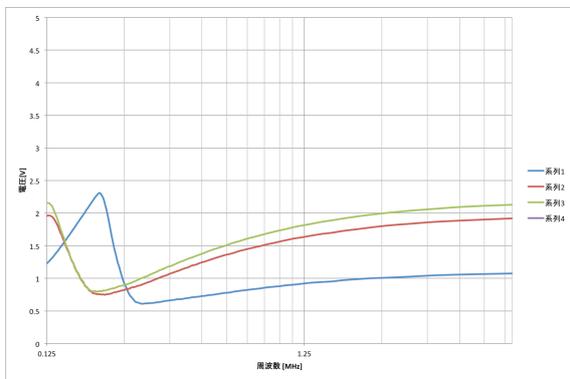
図 4.8(b), 図 4.8(c), 図 4.8(d) は各素子を個別に変更した時の波形である。C2 を小さくすることで指の数による電圧の変化がやや大きくなった。C1 を大きくすることで波形が滑らかになった。R2 を変更することで電圧が底上げされた。図 4.8(e) は C1, C2, R2 の全ての素子を変更した時の波形である。回路素子変更前に比べ、指の数による電圧の変化が大きくなり、触り方を検出することが可能になった。図 4.9 は回路素子改良後のセンシング回路の回路図である。



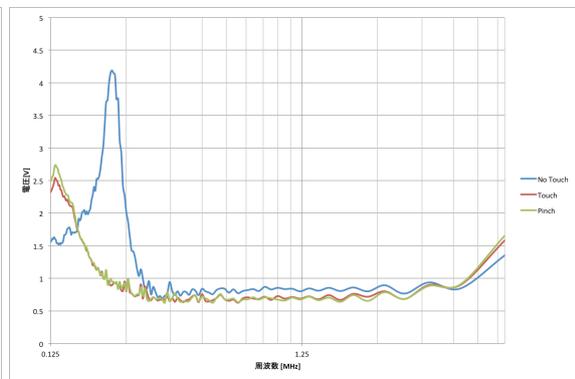
(a) 回路素子改良前



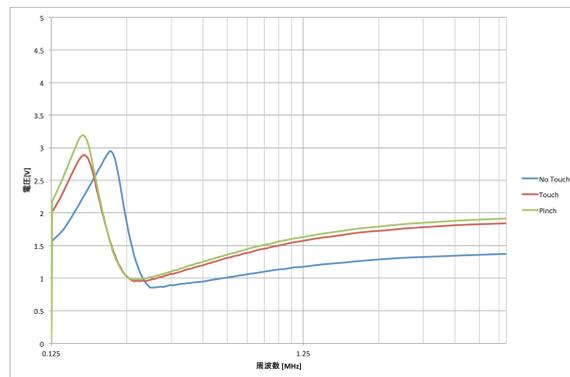
(b) C2: 10nF → 100pF



(c) C1: 100pF → 10nF

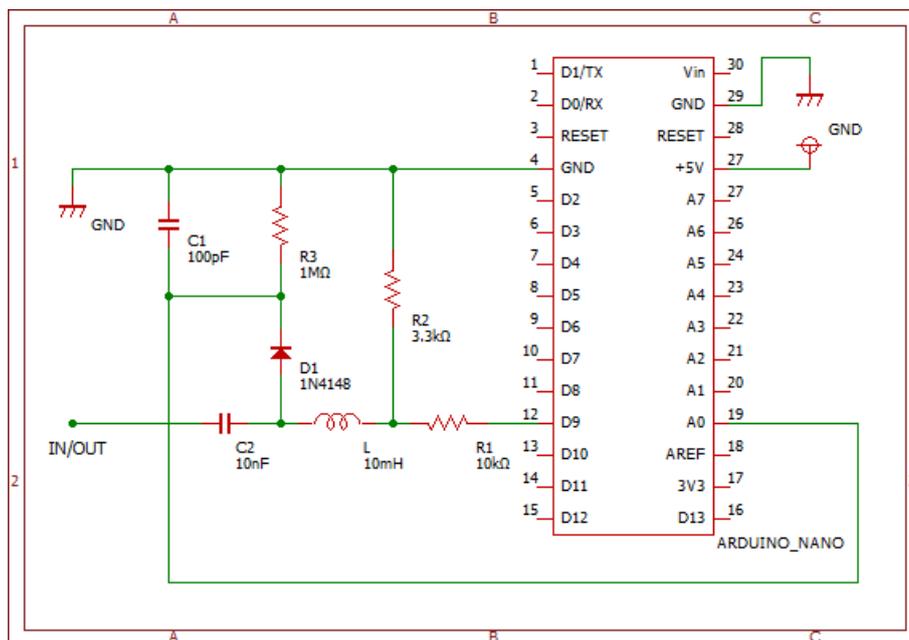


(d) R2: 3.3kΩ → 5kΩ

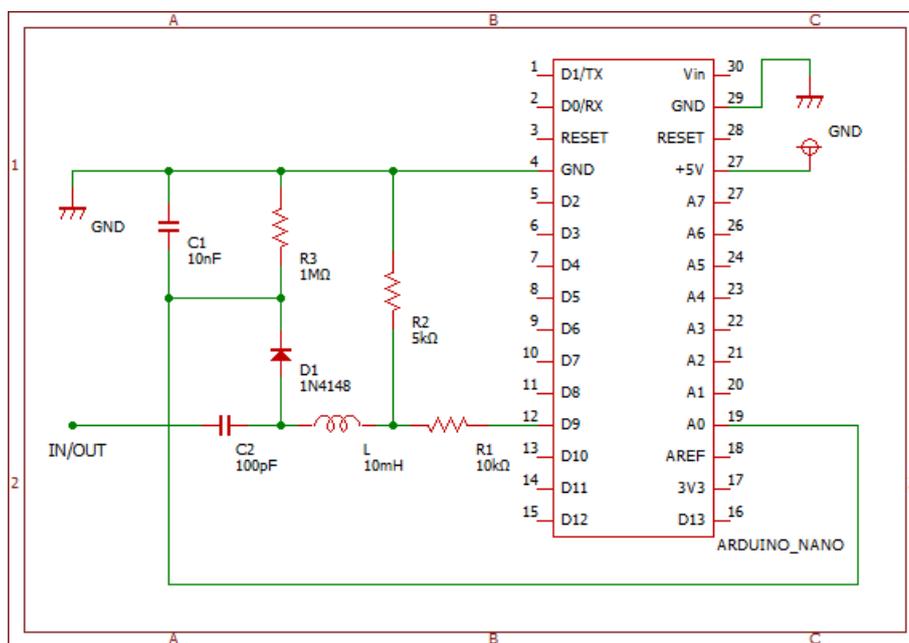


(e) C1: 100pF → 10nF

図 4.8: 回路素子の違いによるセンシングデータの比較



(a) 回路素子改良前



(b) 回路素子改良後

図 4.9: センシング回路の回路図

4.5 触り方の登録と認識

図 4.10 は触り方の登録及び識別を行うアプリケーションである。触り方の識別は SVM(サポートベクターマシン)で行う。(1) は入力波形の描画エリアであり、各周波数において Arduino に入力された電圧が波形として表示されている。先述の通り、この波形は接続している物体及び、触り方によって変化する。この時、ユーザがある触り方をした場合に (2) の対応するボタンを押すことでその触り方の場合のデータを csv ファイルへと出力する。データの形式は先頭にクラス、その後に 127 個のデータという形になる。クラスは (2) のボタンの上から順に 0, 1, 2, 3 に対応する。(3) の FileRead ボタンを押すと出力した csv ファイルのデータを元に教師データを作成する。この時、(4) の各チェックボックスにチェックが入っている触り方のみ教師データを作成し、識別する触り方として登録する。例えば、"No Touch", "Touch", "Pinch", "Grasp" の全ての触り方のデータが csv ファイルに出力されていても、チェックボックスにチェックが入っているものが "No Touch", "Touch", "Grasp" の 3 種類であった場合には、この 3 種類のみ教師データを作成し識別する。従って、この場合どんな触り方をしても "Pinch" とは識別されない。チェックボックスはクリックで任意に変更可能である。ファイルの読み込みを行った際に、登録済みの触り方の場合には対応するボタンの左側に "T" が表示され、未登録の場合には "F" が表示される。触り方の識別結果は (5) に表示される。触り方が一つも登録されていない場合には "No Data" と表示される。(6) の FileClear ボタンを押すとデータが学習したデータが保存されている csv ファイルを空にする。

認識精度を向上させるため、触り方の識別は一定回数センシングを行った際に最も識別回数が多いものを結果とする多数決方式を採用した。今回の実装では 5 回のセンシングを 1 サイクルとして、識別された回数が最も多い触り方を識別結果とした。この識別結果に応じて通信を行う。この際、常に同じ触り方をしていた場合、通信が完了しても再度通信を開始してしまうため、別の触り方から変化した場合に通信を開始するように設定した。

図 4.11 はユーザがスマートオブジェクトに触れることにより、システムが "Touch" と検出した時の画面である。図 4.10 と比較すると波形が変化していることが見て取れる。"Touch" を検出したため、図 4.10 の (5) で "No Touch" と表示されていたものが "Touch" に変化している。

図 4.10, 図 4.11 の描画エリアにおいて縦軸は最小値 0, 最大値 1024, 横軸は最小値 0, 最大値 126 となっている。この縦軸は電圧を示しており、最大値 1024 は電圧 5V に対応している。横軸は周波数を示しており、横軸の値は式 4.1 の x である。すなわち最小値 0 は 8MHz, 最大値 126 は 125kHz に対応している。図 4.12 は縦軸の単位を V, 横軸の単位を MHz に変換し、対数表示させ、グラフもデータの点をプロットして表示するようにしたアプリケーションである。図 4.13 は "Touch" を検出した場合である。このように軸を変換しても同じように動作するが、元の軸の方が波形の変化が視覚的に分かりやすく感じたため、縦軸は最小値 0, 最大値 1024, 横軸は最小値 0, 最大値 126 のものを利用することとした。

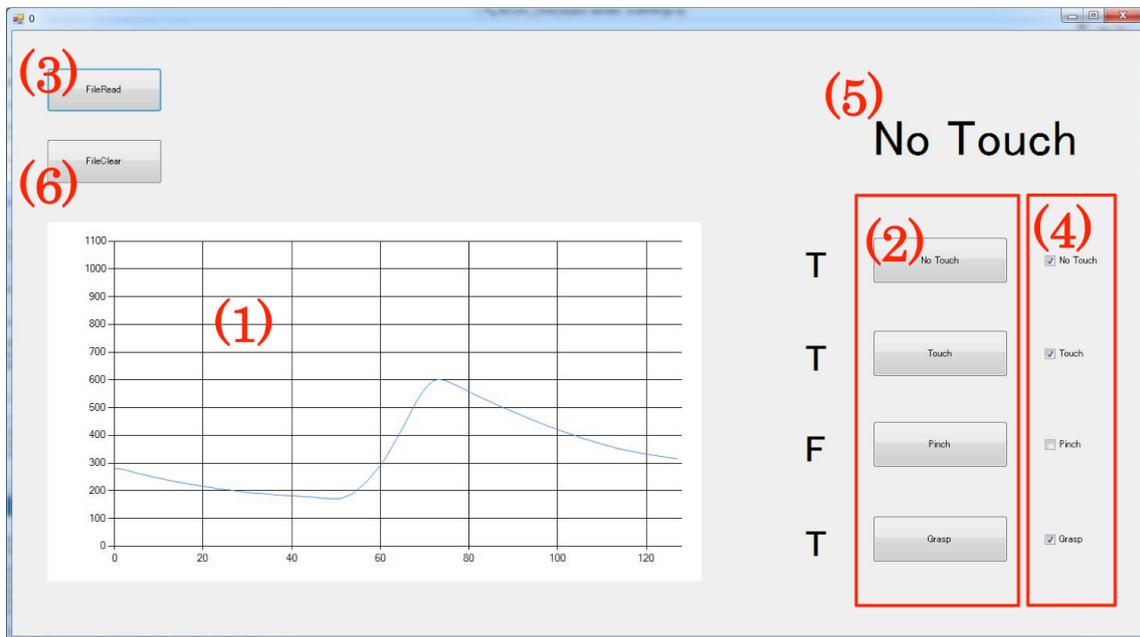


図 4.10: 触り方の登録及び識別を行うアプリケーション

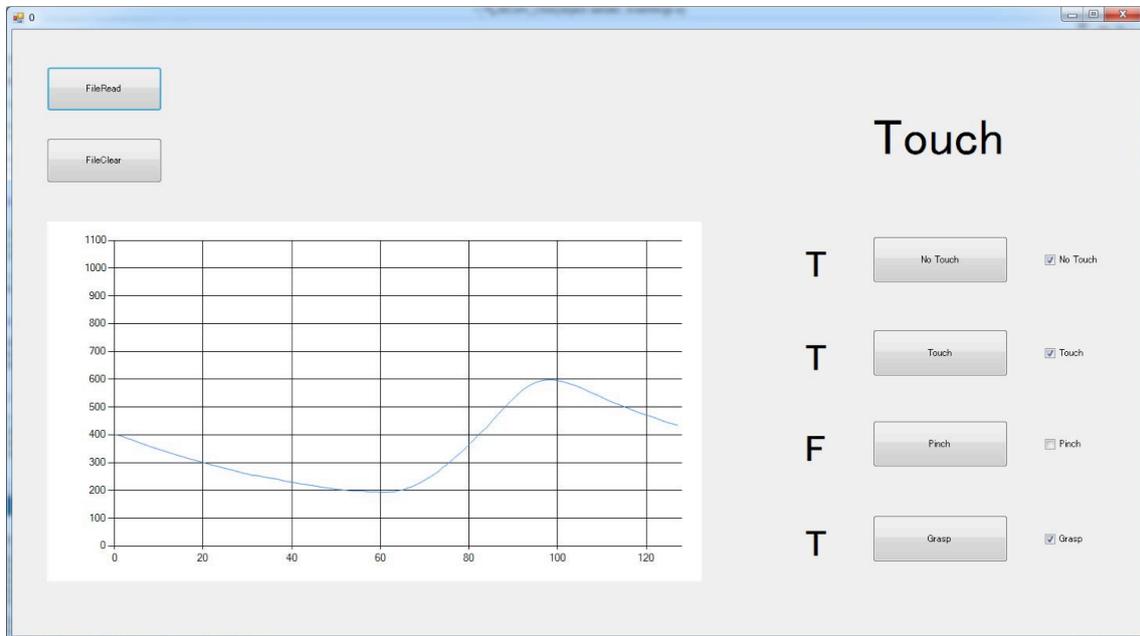


図 4.11: 触り方の登録及び識別を行うアプリケーション (Touch を認識した場合)

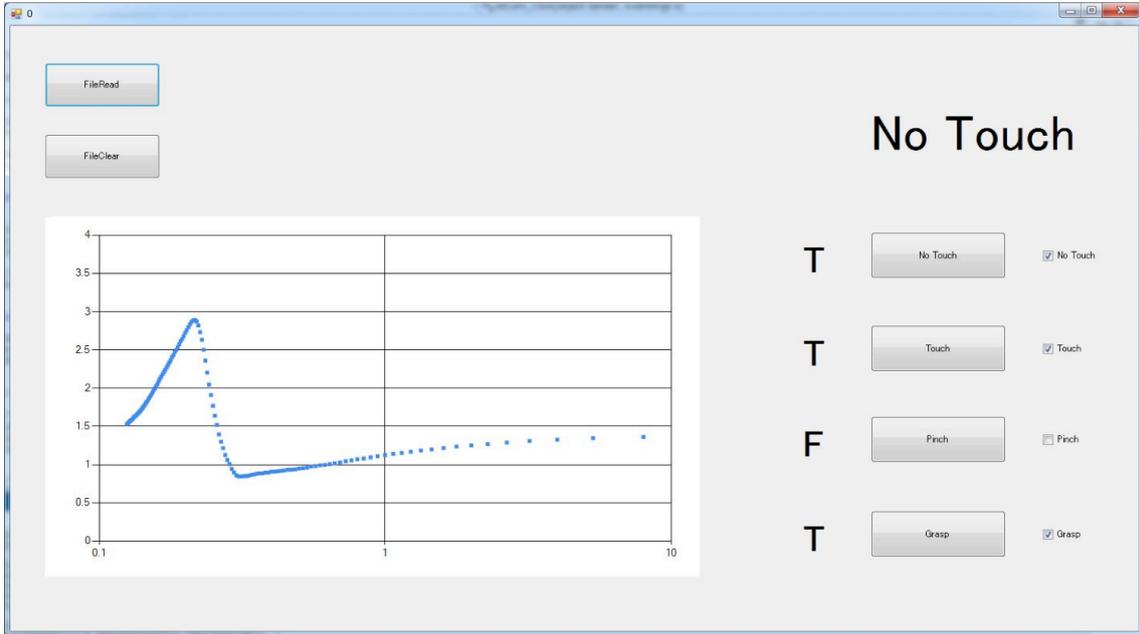


図 4.12: 軸を変換したアプリケーション

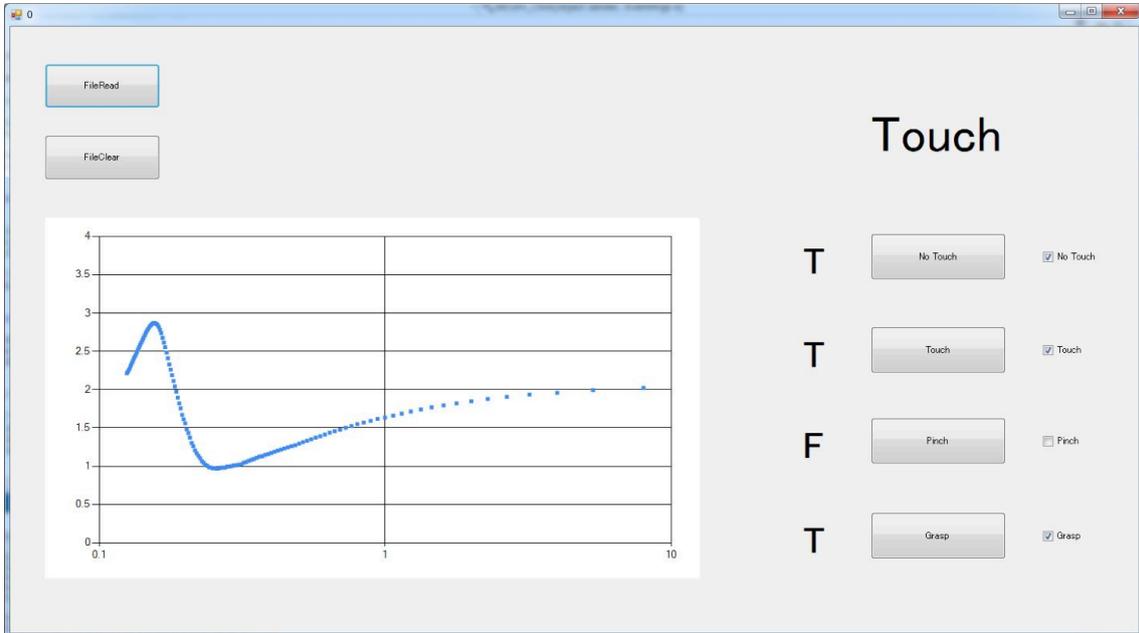


図 4.13: 軸を変換したアプリケーション (Touch を認識した場合)

4.6 回路の切り替え

ウェアラブルデバイスは受信, 送信の2機能を有し, スマートオブジェクトは受信, 送信, センシングの3機能を有する. 各機能を使用する場合には各回路をON, それ以外の回路をOFFにする. このON/OFFは各回路に接続されたリレーのON/OFFを切り替えることで行う. 受信回路のリレーがON, それ以外の回路がOFFになった状態を受信状態, 送信回路のリレーがON, それ以外の回路がOFFになった状態を送信状態, センシング回路のリレーがON, それ以外の回路がOFFになった状態をセンシング状態とする. 従って, ウェアラブルデバイスには受信状態と送信状態の2状態が存在し, スマートオブジェクトには受信状態, 送信状態とセンシング状態の3状態が存在する.

リレーのON/OFFは各Arduinoに接続されたスマートフォン, PCからの命令によって制御する. 回路切り替えの命令はコネクションパケット, スタートパケット, エンドパケットの3種類存在する. 各パケットの具体的な中身は表4.1のような32文字のパケットになっている. Arduinoは接続されたスマートフォン, PCから送られてきたデータが何であるかを識別する. このパケットを利用して回路の切り替えを行う.

表 4.1: 回路切り替えに用いるパケット

パケット名	パケットの中身
コネクションパケット	nttttttttttttttttttttttttttttttt
スタートパケット	syyyyyyyyyyyyyyyyyyyyyyyyyyyyyys
エンドパケット	fiiiiiiiiiiiiiiiiiiiiiiiiif

4.7 システム処理の流れ

システム処理の流れは図4.14のようになる. 初期状態ではスマートオブジェクトではセンシング状態, ウェアラブルデバイスは受信状態となっている. ユーザがスマートオブジェクトに触れると予め登録しておいた教師データにより触り方の識別を行う. その結果, 通信を行うような触り方であった場合, コネクション確立ステップに移る. コネクション確立ステップではスマートオブジェクトからコネクションパケットを送る. この時, スマートオブジェクトのPCからPCに接続されたArduinoへとコネクションパケットが送信されることになる. 区別のため, PCに接続されたArduinoをArduino(A)とする. Arduino(A)がコネクションパケットを受信すると各回路のリレーを制御し送信状態へと切り替える. 送信状態へ切り替わるとArduino(A)はPCから受信したコネクションパケットを, 人体を経由してウェアラブルデバイスへ送る. Arduino(A)はコネクションパケットを送信し終わると受信状態へと切り替える.

ウェアラブルデバイスはこのコネクションパケットを受信するとスマートオブジェクトへコネクションパケットを送り返す. この時, ウェアラブルデバイスのスマートフォンからス

スマートフォンに接続された Arduino へと接続パッケージが送信されることになる。スマートフォンに接続された Arduino を Arduino(B) とする。Arduino(B) が接続パッケージを受信すると各回路のリレーを制御し送信状態へと切り替える。送信状態へ切り替わると Arduino(B) はスマートフォンから受信した接続パッケージを、人体を經由してスマートオブジェクトへ送る。Arduino(B) は接続パッケージを送信し終わると受信状態へと切り替える。スマートオブジェクトがウェアラブルデバイスから送り返された接続パッケージを受信すると、スマートオブジェクトとウェアラブルデバイスは接続が確立されたとして、データ通信ステップへと移行する。

スマートオブジェクトから接続パッケージを送信した際、接続パッケージがウェアラブルデバイスへ届かない、あるいはウェアラブルデバイスは接続パッケージを受信したが、送り返した接続パッケージがスマートオブジェクトに届かないことにより、接続の確立が失敗する可能性がある。この場合でも接続の確立が行えるようにするため、スマートオブジェクトが接続パッケージを送信してから 100ms 以内に接続パッケージが送り返されてこない場合、接続パッケージを再送する。これにより、一度接続の確立が失敗した場合でも再度接続の確立が行えるようにする。接続パッケージの再送は最大で 50 回行うように設定した。50 回再送を行っても接続の確立が失敗する場合、ユーザは再度スマートオブジェクトを触り直して接続確立をやり直すことになる。

データ通信ステップに移行するとスマートオブジェクトはスタートパッケージを送信する。Arduino(A) がスタートパッケージを受信すると再び送信状態へと切り替える。このスタートパッケージはウェアラブルデバイスへは送信しない。スマートオブジェクトはスタートパッケージを送信した後、目的のデータを送信する。送信状態となったため、このデータはウェアラブルデバイスへと送信される。スマートオブジェクトは目的のデータを送信し終わると通信終了ステップへと移行する。

通信ステップに移行するとスマートオブジェクトはエンドパッケージを送信する。Arduino(A) がエンドパッケージを受信すると初期状態のセンシング状態へと切り替え、通信を終了する。これにより、スマートオブジェクトとウェアラブルデバイスともに初期状態へと戻る。以降は同様の動作を続けていく。

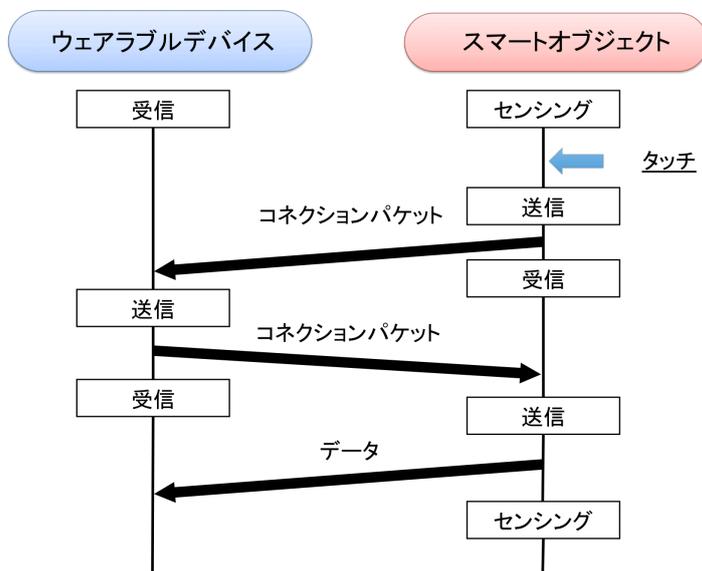


図 4.14: システム処理の流れ

4.8 アプリケーション：触り方に応じた画像通信

触り方を検出可能な人体通信システムを利用し、通信する画像を触り方によって選択するアプリケーションを作成した。スマートオブジェクトに触り方を登録しておき、触り方に応じて通信する画像を対応付けておく。スマートオブジェクトでのセンシングにより、登録しておいた触り方を検出する対応付けた画像を通信する。複数の触り方を登録しておくことで、触り方によって通信する画像を選択することができる。

4.8.1 スマートオブジェクトのアプリケーション

図 4.15 はスマートオブジェクトのアプリケーションである。開発環境として Microsoft Visual Studio Ultimate 2012 を使用し、開発言語には C# を使用した。(1) は入力波形の描画エリアであり、各周波数において Arduino に入力された電圧が波形として表示されている。(2) のボタンを押すことでその触り方の場合のデータを csv ファイルへと出力する。(3) の FileRead ボタンを押すと出力した csv ファイルのデータを元に教師データを作成する。(4) の各チェックボックスにチェックが入っている触り方のみ教師データを作成し、識別する触り方として登録する。触り方の識別結果は (5) に表示される。触り方が一つも登録されていない場合には "No Data" と表示される。(6) の FileClear ボタンを押すとデータが学習したデータが保存されている csv ファイルを空にする。(7) が通信する通信する画像である。左から順に (2) の二つ目のボタンに割り当てられている触り方と対応付けられている。例えば、2 つ目のボタンに割り当てた "Touch" という触り方を検出した場合、一番左の画像を通信し、3 つ目のボタンに割り当てた "Pinch" という触り方を検出した場合、真ん中の画像を通信する。(2) の 1 つ目のボタンは定常状態として何も触らない "No Touch" を割り当てるものとし、この場合には画像の通信は行わない。

4.8.2 ウェアラブルデバイスのアプリケーション

図 4.16、図 4.17 はウェアラブルデバイスのアプリケーション画面である。開発環境として Eclipse 4.3 Kepler を使用し、開発言語には Java (JDK 1.7.0 45) を使用した。平常時にはギャラリーとして複数の画像が表示されている。ユーザがスマートオブジェクトに触れると画像の通信を行い、通信が正しく完了すると送られてきた画像がディスプレイに表示される。

4.8.3 データ通信の仕組み

先述のシステム処理の流れに従い、コネクションを確立すると、触り方に応じたデータの通信を開始する。初めに画像のデータがバイト型の配列に変換される。この配列の大きさがデータのファイルサイズとなる。通信の際にはまずこのファイルサイズを送信する。このファイルサイズは 8 バイトのバイト型配列に変換され、シリアル通信により 1 ビットずつ Arduino へ送られる。Arduino はテーブルトップ PC からデータが送られて来たらそのデータを受信し、

その都度信号をウェアラブルデバイスへ向けて出力する。ただし、Arduino の処理が PC よりも遅れ、それが重なっていくと Arduino の受信バッファが溢れデータが消失してしまう。今回使用した Arduino Nano 3.1 では受信バッファは 64 バイトまでしか蓄積できなかったため、PC は 64 バイト送信する毎に 20ms 待機させるようにした。これにより、Arduino の受信バッファが溢れることなくデータが送れるようになっている。通信速度の理論値は 17.7kbps となる。

ウェアラブルデバイスでは初めにファイルサイズとして 8 バイトのデータを受信する。これがもし 8 バイトとなっていなければノイズであるとしてその後の処理を行わない。8 バイトのデータを受信すれば、これがファイルサイズのデータであると認識して数値に変換し、その後もデータの受信を行う。正常に通信が行われていればそのままデータがバイト型配列として送られてくる。このとき送られてくるデータサイズは初めに受信したファイルサイズの分のはずである。そのため、受信したデータのバイト配列の大きさが初めに受信したファイルサイズと一致していれば、正しい画像情報であるとしてバイト型配列を元のデータに復元する。

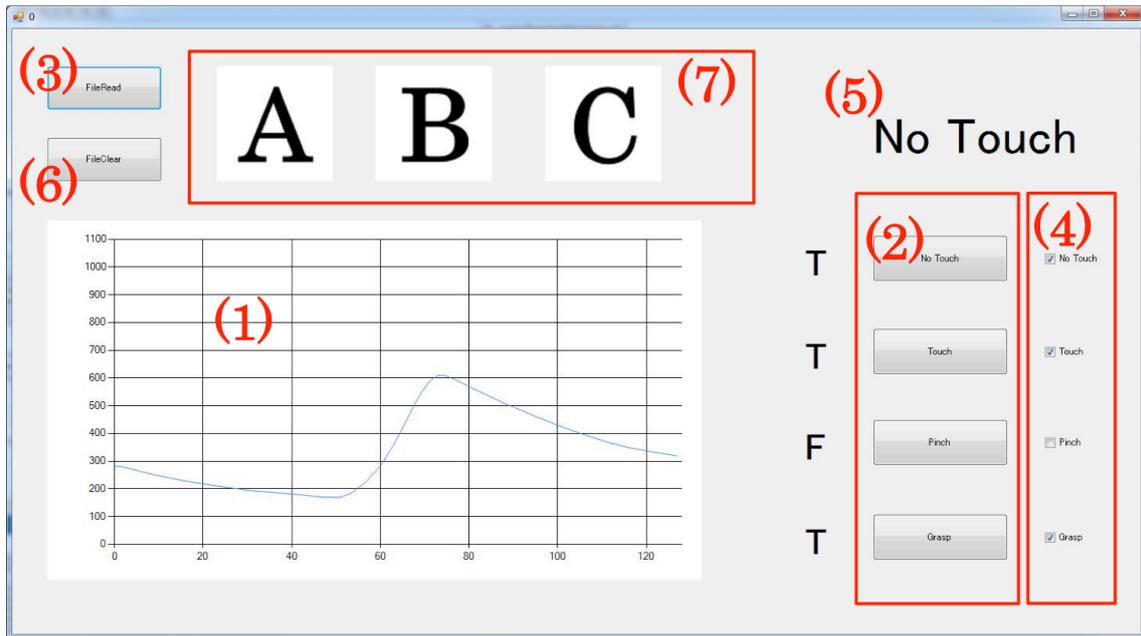
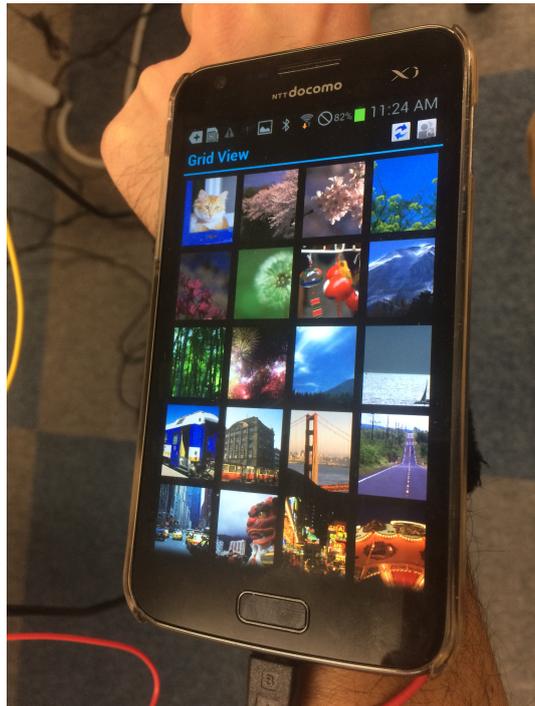
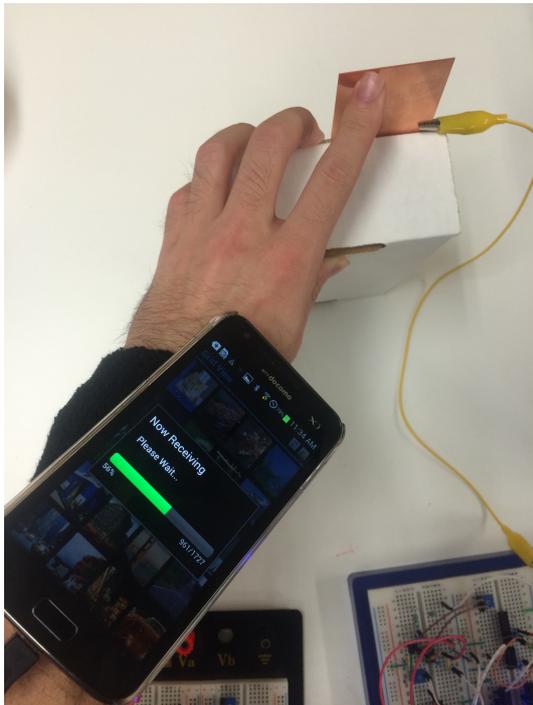


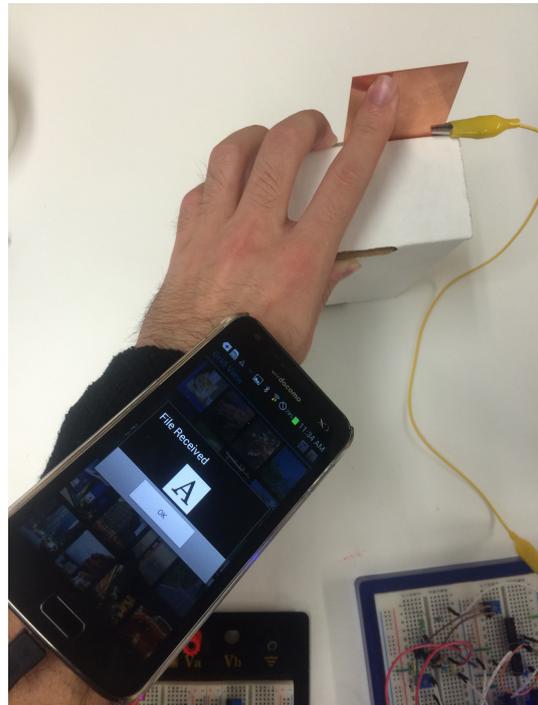
図 4.15: スマートオブジェクトのアプリケーション



(a) 待機画面

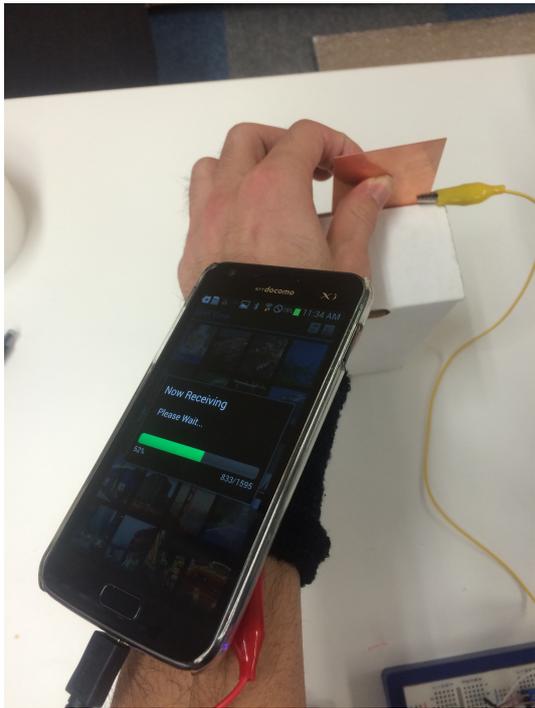


(b) 通信中 (画像 A)

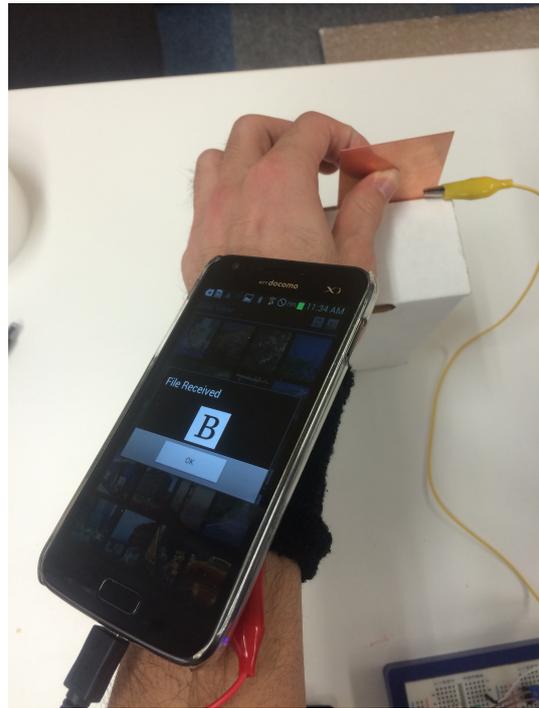


(c) 通信完了 (画像 A)

図 4.16: ウェアラブルデバイスのアプリケーション (1/2)



(a) 通信中 (画像 B))



(b) 通信完了 (画像 B)

図 4.17: ウェアラブルデバイスのアプリケーション (2/2)

第5章 評価

5.1 実験 1: 素材の違いによるセンシングデータの測定

本研究では導体にセンサを取り付けることでセンシングを行い，触り方を検出する．その際，導体の素材によりセンシングデータに違いが出るかを測定した．

5.1.1 被験者

著者が被験者として実験を行った．

5.1.2 実験内容

素材として銅，真鍮，アルミニウム，ステンレスの4種類の金属板を用意した．金属板は全て $0.5 \times 50 \times 100$ [mm] である．各物質の抵抗率を表 5.1 に示す [27]．図 5.1 は用意した金属板である．左から順に真鍮，アルミニウム，ステンレス，銅である．図 5.2 は実験環境である．触り方は，図 5.3 のように何も触れない”No Touch”，1本の指で触れる”Touch”，2本の指で摘む”Pinch”，5本全ての指で握る”Grasp”の4種類とした．この時，金属板には左手で触れるようにした．各触り方をした際のセンシングデータを取得した．

表 5.1: 各物質の抵抗率

物質	体積固有抵抗率 (20 °C)[$\mu\Omega \cdot \text{cm}$]	温度係数 [$1/^\circ\text{C}$]
銅	1.72	4.33×10^{-3}
真鍮 (黄銅)	6.2	4.1×10^{-3}
アルミニウム	2.75	4.2×10^{-3}
ステンレス	60	2×10^{-3}



図 5.1: 実験で使用した金属板

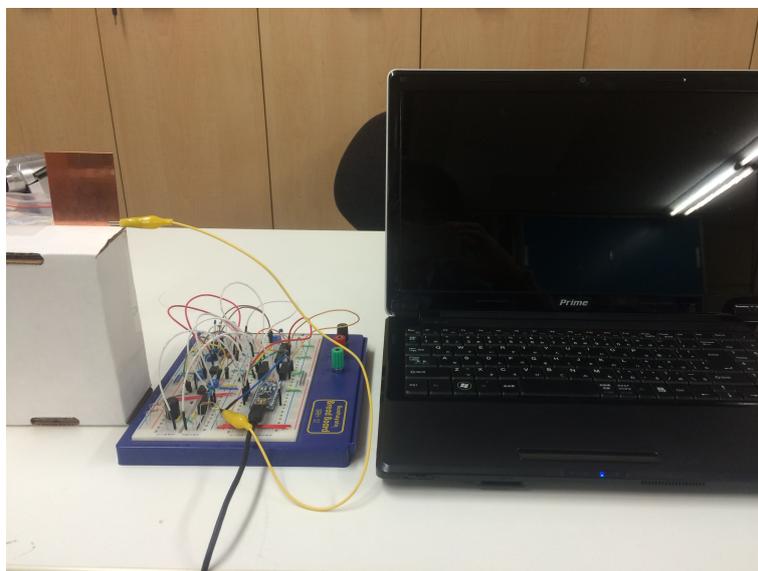
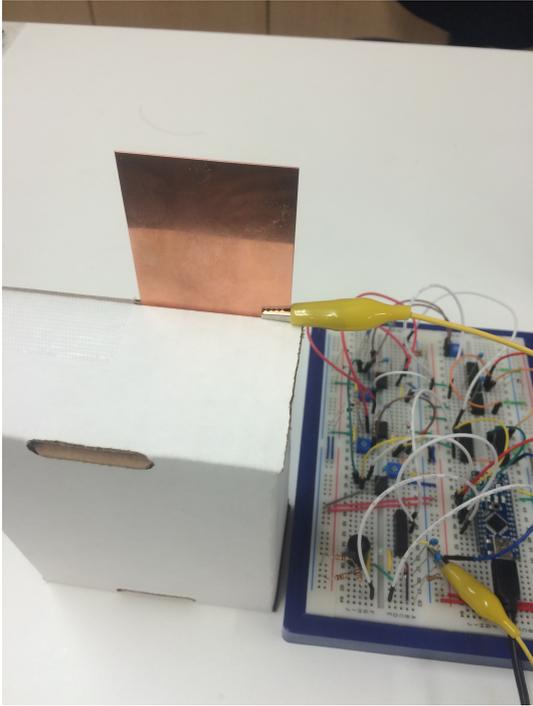
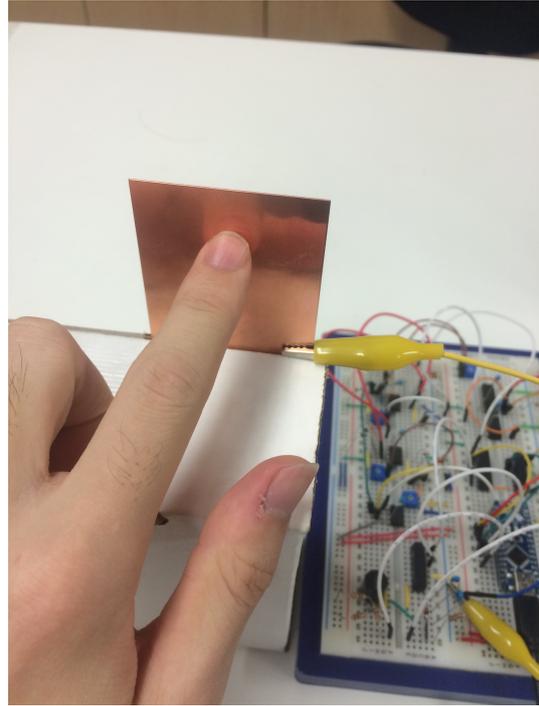


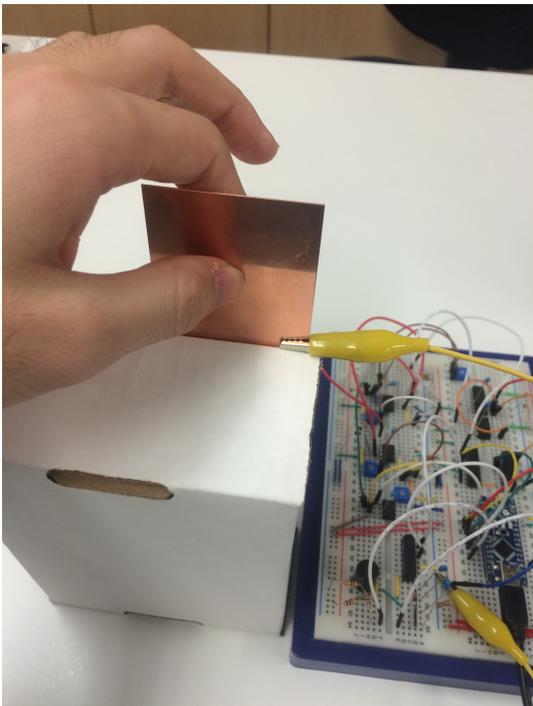
図 5.2: 実験環境



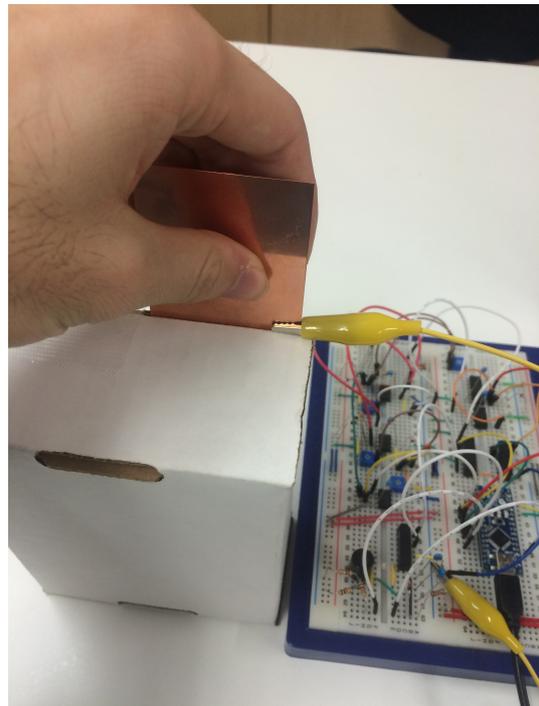
(a) No Touch



(b) Touch



(c) Pinch



(d) Grasp

図 5.3: 各触り方 (銅板)

5.1.3 実験結果

図 5.4 は各素材毎の触り方の違いによるセンシングデータの比較結果である。どの素材の場合も”No Touch”とそれ以外には明確な違いが生じた。被験者が素材に触れた場合の3種類の触り方を比較すると、”Touch”とその他2つには差が生じている。一方で”Pinch”と”Grasp”を比較した場合、”Grasp”の方が波形のピーク値が高くなったが、大きな差は生じなかった。

図 5.5 は各触り方毎の素材の違いによるセンシングデータの比較結果である。どの触り方をした場合でも4種類の波形はほぼ重なり、大きな差は生じなかった。

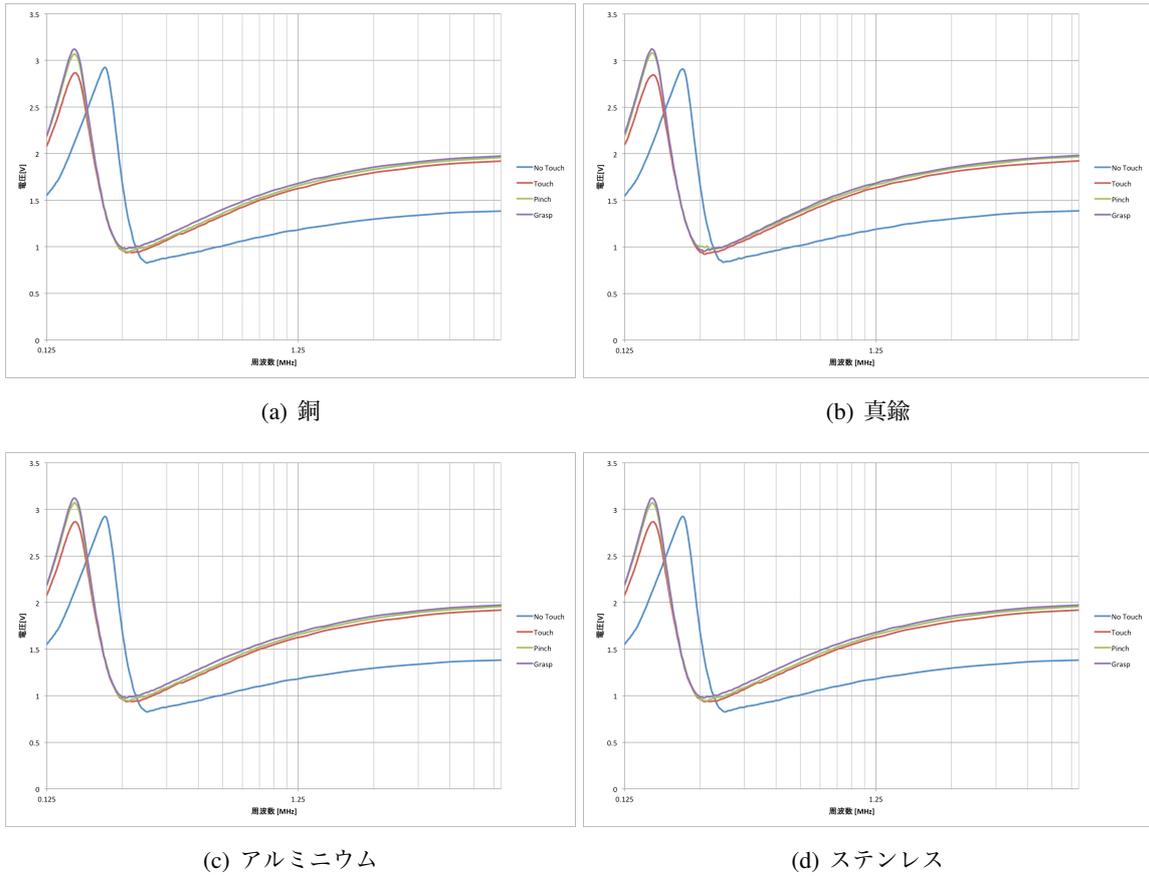
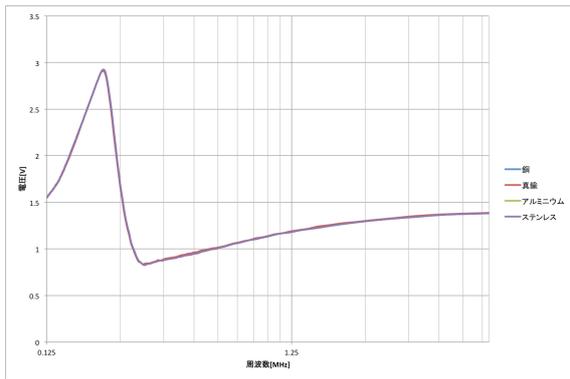
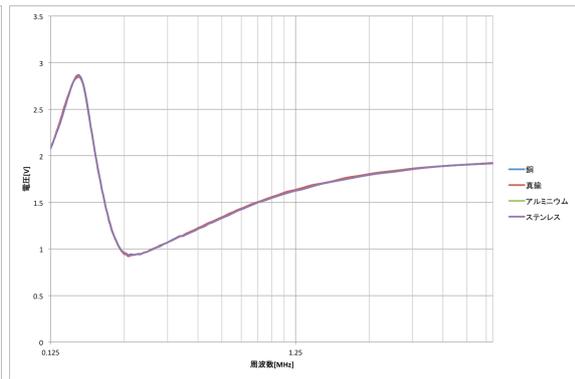


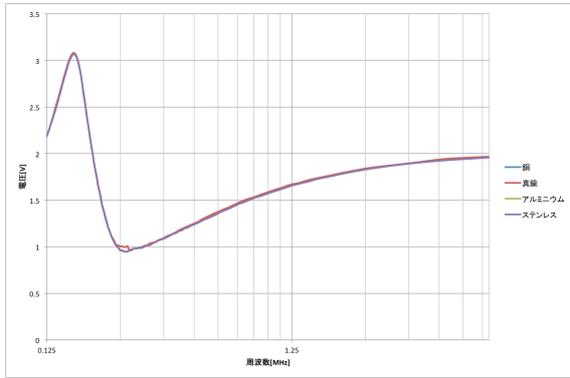
図 5.4: 各素材毎の触り方の違いによるセンシングデータの比較



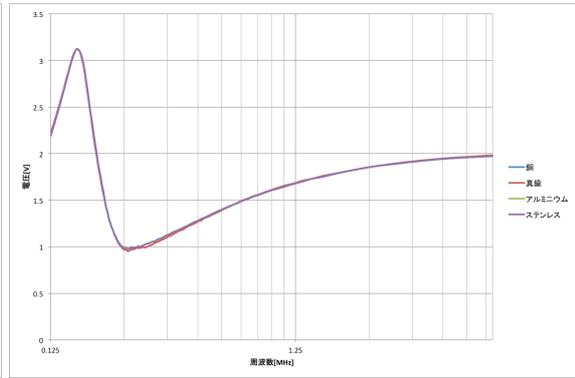
(a) No Touch



(b) Touch



(c) Pinch



(d) Grasp

図 5.5: 各触り方毎の素材の違いによるセンシングデータの比較

5.2 実験 2:触り方検出の認識率

スマートオブジェクトへ触れた際に触り方の認識率を測定する。

5.2.1 被験者

被験者は 22 歳～25 歳の大学生・大学院生 4 名である。

5.2.2 実験内容

スマートオブジェクトとして実験 1 で使用した 0.5 x 50 x 100[mm] の銅板及び直方体のプラスチックの周りを銅箔で包んだ導体の物体であるスマートボックスの 2 種類を用意した。スマートオブジェクトへの触り方は図 5.3 及び図 5.6 のように何も触れない”No Touch”，1 本の指で触れる”Touch”，2 本の指で摘む”Pinch”，5 本全ての指で握る”Grasp”の 4 種類とした。スマートオブジェクトへは左手で触れるよう指示した。学習データは被験者毎に取得し、各触り方において 10 回分のデータを取得した。実験は識別する触り方を”No Touch”，”Touch”，”Pinch”の 3 種類，”No Touch”，”Touch”，”Grasp”の 3 種類，”No Touch”，”Pinch”，”Grasp”の 3 種類，”No Touch”，”Touch”，”Pinch”，”Grasp”の 4 種類とした場合でそれぞれ実施した。例えば，”No Touch”，”Touch”，”Pinch”の場合は”Grasp”は学習データに含めず，”Grasp”と識別されることもない。

5.2.3 実験手順

被験者は以下の手順の通りに実験を行う。

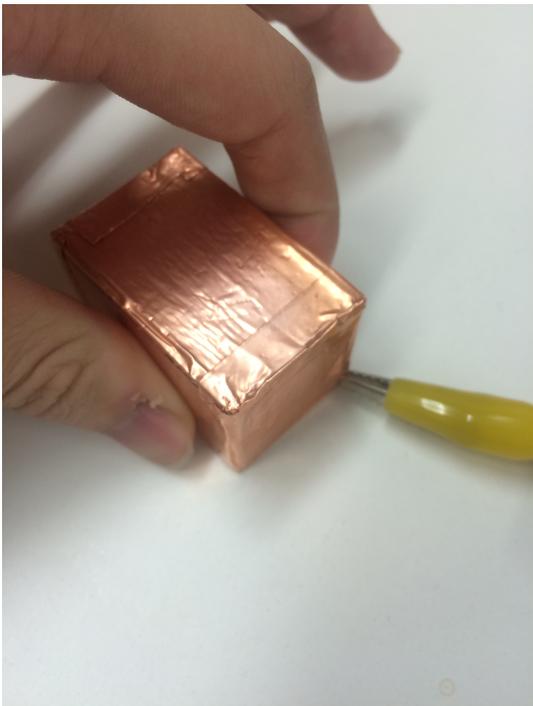
1. 各触り方がどのようなものか説明を受ける。
2. 各触り方の学習データを作成する。
3. 各触り方を 5 回行う。



(a) No Touch



(b) Touch



(c) Pinch



(d) Grasp

図 5.6: 各触り方 (スマートボックス)

5.2.4 実験結果

表 5.2, 表 5.3, 表 5.4, 表 5.5, 表 5.6 は銅板に対する触り方の識別結果である。表の見方は行の触り方を行った場合に、システムが列の触り方と認識した割合をパーセンテージで表している。例えば、表 5.6 は”No Touch”の場合に”No Touch”と識別されたのが 100%, ”Touch”の識別されたのが 0%, ”Pinch”と識別されたのが 0%, ”Grasp”と識別されたのが 0%, ”Touch”を行った場合に”No Touch”と識別されたのが 0%, ”Touch”と識別されたのが 100%, ”Pinch”と識別されたのが 0%, ”Grasp”と識別されたのが 0%, ”Pinch”を行った場合に”No Touch”と識別されたのが 0%, ”Touch”と識別されたのが 0%, ”Pinch”と識別されたのが 75%, ”Grasp”と識別されたのが 25%, ”Grasp”を行った場合に”No Touch”と識別されたのが 0%, ”Touch”の識別されたのが 0%, ”Pinch”と識別されたのが 5%, ”Grasp”と識別されたのが 95%という結果を表している。

表 5.7, 表 5.8, 表 5.9, 表 5.10, 表 5.11 はスマートボックスに対する触り方の識別結果である。表の見方は銅板の場合と同様である。

実験結果から、概ね 4 種類の触り方は識別できるものの、”Pinch”と”Grasp”は誤認識される場合も見られた。特に個人によっては”Pinch”と”Grasp”は認識精度が著しく低い結果が生じた。一方で、”No Touch”の時に他の”触れた”と識別されたことは一度も発生しなかった。また、”Touch”の時に”Pinch”や”Grasp”と識別されたことも一度も発生しなかった。この実験結果から、個人毎に学習データを取得した場合、”触れていない”か”1本の指で触れた”か”2本以上の指で触れた”かという 3 パターンは高い精度で識別可能であると言える。

表 5.2: 銅板に対する触り方の識別結果 (被験者 A)

(a) No Touch, Touch, Pinch

被験者 A	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

被験者 A	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

被験者 A	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	100	0
Grasp	0	0	100

(d) No Touch, Touch, Pinch, Grasp

被験者 A	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	100	0
Grasp	0	0	0	100

表 5.3: 銅板に対する触り方の識別結果 (被験者 B)

(a) No Touch, Touch, Pinch

被験者 B	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

被験者 B	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

被験者 B	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	0	100
Grasp	0	0	100

(d) No Touch, Touch, Pinch, Grasp

被験者 B	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	0	100
Grasp	0	0	0	100

表 5.4: 銅板に対する触り方の識別結果 (被験者 C)

(a) No Touch, Touch, Pinch

被験者 C	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

被験者 C	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

被験者 C	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	100	0
Grasp	0	0	100

(d) No Touch, Touch, Pinch, Grasp

被験者 C	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	100	0
Grasp	0	0	0	100

表 5.5: 銅板に対する触り方の識別結果 (被験者 D)

(a) No Touch, Touch, Pinch

被験者 D	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

被験者 D	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

被験者 D	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	100	0
Grasp	0	60	40

(d) No Touch, Touch, Pinch, Grasp

被験者 D	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	100	0
Grasp	0	0	20	80

表 5.6: 銅板に対する触り方の識別結果 (全体の平均)

(a) No Touch, Touch, Pinch

全体	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

全体	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

全体	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	75	25
Grasp	0	15	85

(d) No Touch, Touch, Pinch, Grasp

全体	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	75	25
Grasp	0	0	5	95

表 5.7: スマートボックスに対する触り方の識別結果 (被験者 A)

(a) No Touch, Touch, Pinch

被験者 A	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

被験者 A	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

被験者 A	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	100	0
Grasp	0	0	100

(d) No Touch, Touch, Pinch, Grasp

被験者 A	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	100	0
Grasp	0	0	0	100

表 5.8: スマートボックスに対する触り方の識別結果 (被験者 B)

(a) No Touch, Touch, Pinch

被験者 B	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

被験者 B	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

被験者 B	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	100	0
Grasp	0	0	100

(d) No Touch, Touch, Pinch, Grasp

被験者 B	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	100	0
Grasp	0	0	20	80

表 5.9: スマートボックスに対する触り方の識別結果 (被験者 C)

(a) No Touch, Touch, Pinch

被験者 C	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

被験者 C	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

被験者 C	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	80	20
Grasp	0	40	60

(d) No Touch, Touch, Pinch, Grasp

被験者 C	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	100	0
Grasp	0	0	20	80

表 5.10: スマートボックスに対する触り方の識別結果 (被験者 D)

(a) No Touch, Touch, Pinch

被験者 D	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

被験者 D	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

被験者 D	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	80	20
Grasp	0	0	100

(d) No Touch, Touch, Pinch, Grasp

被験者 D	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	60	40
Grasp	0	0	20	100

表 5.11: スマートボックスに対する触り方の識別結果 (全体の平均)

(a) No Touch, Touch, Pinch

全体	No Touch	Touch	Pinch
No Touch	100	0	0
Touch	0	100	0
Pinch	0	0	100

(b) No Touch, Touch, Pinch

全体	No Touch	Touch	Grasp
No Touch	100	0	0
Touch	0	100	0
Grasp	0	0	100

(c) No Touch, Pinch, Grasp

全体	No Touch	Pinch	Grasp
No Touch	100	0	0
Pinch	0	90	10
Grasp	0	35	65

(d) No Touch, Touch, Pinch, Grasp

全体	No Touch	Touch	Pinch	Grasp
No Touch	100	0	0	0
Touch	0	100	0	0
Pinch	0	0	90	10
Grasp	0	0	10	90

5.3 実験 3:システム利用

本システムをユーザに使用してもらい、触り方の検出からデータ通信が完了するまでのタスク成功率を測定した。

5.3.1 被験者

被験者は 22 歳～24 歳の大学生・大学院生 4 名である。

5.3.2 実験内容

スマートオブジェクトとして実験 2 で使用したスマートボックスを用意した。スマートオブジェクトへの触り方は実験 2 の結果より識別精度の高かった、何も触れない”No Touch”，1 本の指で触れる”Touch”，5 本全ての指で握る”Grasp”の 3 種類とした。 ”Touch”あるいは”Grasp”が検出されるとそれぞれの触り方に対応した画像の通信を行う。被験者は右手にウェアラブルデバイスを装着し、スマートオブジェクトへは左手で触れるよう指示した。 ”Touch”が検出された場合に通信する画像 A は 1.72KB， ”Grasp”が検出された場合に通信する画像 B は 1.59KB である。被験者は各触り方を行い、触り方に応じた画像がウェアラブルデバイスで受信できるか実験する。学習データは被験者毎に取得し、各触り方において 10 回分のデータを取得した。初めにこの条件で実験を行ったところ、ウェアラブルデバイスで受信したデータには欠損が生じ、正常にデータを受信することができず、タスクを完了することができなかった。そのため、タスクが完了できるよう通信を安定させるためにスマートオブジェクトとウェアラブルデバイスの GND を接続した状態で実験を行った。

5.3.3 実験手順

被験者は以下の手順の通りに実験を行う。

1. 各触り方がどのようなものか説明を受ける。
2. 各触り方の学習データを作成する。
3. 各触り方を 5 回行う。

5.3.4 実験結果

表 5.12 はタスク成功率である。被験者 C が”Grasp”の際に失敗しているものの、その他の場合にはタスクが成功している。被験者 C が失敗したのは”Grasp”が”Touch”と識別され、画像 B を受信したものである。この結果より触り方を”触れていない”か”1 本の指で触れた”か”2 本以上の指で触れた”かという 3 パターンとし、スマートオブジェクトとウェアラブルデバイ

スの GND を接続した場合だとシステムは安定して動作すると言える。GND を接続しない場合、受信回路のオープンアンプの倍率を調整することで安定性を改善することができる。しかし、倍率が低いと復元すべき信号が増幅されず、倍率が高いと復元すべきでないノイズまで増幅してしまう。人によって抵抗値等に差があるため、倍率の決定が難しい。GND を接続しない状態での通信の安定化は今後の大きな課題といえる。

表 5.12: タスク成功率

	Touch	Grasp
被験者 A	100	100
被験者 B	100	100
被験者 C	100	80
被験者 D	100	100
平均	100	95

第6章 結論

本研究では身の回りのスマートオブジェクトと簡潔かつ直感的に通信できるシステムを目的とした、触り方を検出可能な人体通信システムを開発した。ユビキタスインタラクションではユーザは使い方を意識することなく、直感的な動作によって通信を行えることが望ましい。物体に触るという行為は人間にとって自然な動作であり、直感的であるといえる。端末間で通信を行うためには端末間の接続設定が必要である。例えば、同じネットワークに接続するという設定やどの端末と接続するかを設定を行う必要がある。これに対し、人体通信では人体に触れている端末同士が人体を通じて直接信号のやりとりを行うことができる。そのため、触れるという行為がそのまま接続設定の代わりとなり、予め接続設定を行う必要がなくなる。また、触り方を検出することで触り方に応じてシステムの動作を変えることができ、触り方に応じたデータを通信することができる。このように、触り方の検出と人体通信を組み合わせることで、触るという直感的な動作のみでユーザの意図を反映したインタラクションを行うことができる。触り方を検出可能な人体通信システムを実現するために、センシング機能と人体通信機能を持つスマートオブジェクトと人体通信機能を持つウェアラブルデバイスを作成した。センシングモジュールでは周波数掃引によって触り方の検出を可能にし、触り方に応じたデータを人体通信により通信できるシステムを構築した。

システムの評価実験から、個人毎に学習データを取得した場合、“触れていない”か“1本の指で触れた”か“2本以上の指で触れた”かという3パターンは高い精度で識別可能であるという結果が得られた。また、スマートオブジェクトとウェアラブルデバイス間のGNDを接続した場合にはシステムは高い精度で動作した。一方で、GNDを接続しない場合、通信は不安定なものとなった。高精度で検出できる触り方の種類を増やす点と、GNDを接続させない状態でシステムを安定して動作させる点は今後の大きな課題であると言える。

謝辞

本論文を執筆するにあたり，指導教員である高橋伸准教授には丁寧なご指導を賜りました。国際会議での発表や論文の執筆を果たすことができたのも，高橋先生のお力添えあつてのことと思っております。改めまして，ここに厚く御礼申し上げます。また，田中二郎教授，三末和男教授，志築文太郎准教授，嵯峨智准教授，Simona Vasilache 助教には，全体ゼミでの発表練習等を通して大変貴重なご意見・ご協力を頂きました。心より御礼申し上げます。

本研究のハードウェアを開発する上で本大学の庄野和宏准教授には，他研究室の所属であるにもかかわらず電気回路の設計について大変多くのご助言を頂きました。この場を借りて厚く御礼申し上げます。

また，インタラクティブプログラミング研究室の皆様にはゼミや日常生活の中で数々のご意見をいただきました。特に，ユビキタスチームの皆様にはゼミ以外にも研究生生活全体にわたって数多くのご意見やご指摘をいただきました。心より御礼申し上げます。そして最後に，大学生活を送る中，経済面や精神面にわたっても支えてくれた両親，そして大学生活を共に過ごし様々な面でお世話になった全ての友人に心より感謝致します。

参考文献

- [1] Hiroshi Ishii and Brygg Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '97*, pp. 234–241, New York, NY, USA, 1997. ACM.
- [2] Tobias Grosse-Puppendahl, Sebastian Herber, Raphael Wimmer, Frank Englert, Sebastian Beck, Julian von Wilmsdorff, Reiner Wichert, and Arjan Kuijper. Capacitive near-field communication for ubiquitous interaction and perception. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14*, pp. 231–242, New York, NY, USA, 2014. ACM.
- [3] Yuto Kondo, Shin Takahashi, and Jiro Tanaka. Information select and transfer between touch panel and wearable devices using human body communication. In *Human-Computer Interaction: Interaction Technologies*, pp. 208–216. Springer, 2015.
- [4] Yuto Kondo, Jiro Tanaka, and Shin Takahashi. Ubi: User body interface for wearable and touch panel device. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, UbiComp/ISWC'15 Adjunct*, pp. 301–304, New York, NY, USA, 2015. ACM.
- [5] T. G. Zimmerman. Personal area networks: Near-field intrabody communication. *IBM Syst. J.*, Vol. 35, No. 3-4, pp. 609–617, September 1996.
- [6] Marc Simon Wegmüller, et al. *Intra-body communication for biomedical sensor networks*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 17323, 2007, 2007.
- [7] Jing-Wu He, Yue-Ming Gao, Sio Hang Pun, Peng Un Mak, and Mang I. Vai. A transceiver designed for intra-body communication of body area networks. In *Proceedings of the 6th International Conference on Body Area Networks, BodyNets '11*, pp. 67–70, ICST, Brussels, Belgium, Belgium, 2011. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [8] 高橋智仁, 夕部彰大, 島田竜也, 岩下克. 人体通信における高周波信号伝搬特性. 電気学会研究会資料. EMC, 電磁環境研究会, Vol. 2010, No. 39, pp. 45–49, 2010.

- [9] 藤井勝之, 伊藤公一, 田島茂. 人体を伝送路として利用したウェアラブル機器の全身モデルを用いた信号伝搬状況の解析. 映像情報メディア学会大会講演予稿集, Vol. 2003, No. 0, pp. 140–140, 2003.
- [10] 菅良太郎, 伊藤公一, 羽賀望, 齊藤一幸, 高橋応明. Mhz 帯における人体近傍通信の基本特性. 電子情報通信学会技術研究報告. A・P, アンテナ・伝播, Vol. 109, No. 339, pp. 41–45, 2009.
- [11] 吉田森之介. 人体通信によるユーザビリティの向上: タッチ式ドアエントリーシステムの開発. Master's thesis, 法政大学大学院, 2005.
- [12] Nobuyuki Matsushita, Shigeru Tajima, Yuji Ayatsuka, and Jun Rekimoto. Wearable key: Device for personalizing nearby environment. In *Wearable Computers, The Fourth International Symposium on*, pp. 119–126. IEEE, 2000.
- [13] Kenji Suzuki, et al. Enhanced touch: a wearable device for social playware. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, p. 83. ACM, 2011.
- [14] Paul Dietz and Darren Leigh. Diamondtouch: A multi-user touch technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pp. 219–226, New York, NY, USA, 2001. ACM.
- [15] Jun Rekimoto. Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 113–120. ACM, 2002.
- [16] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. Touché: Enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 483–492, New York, NY, USA, 2012. ACM.
- [17] Ilya Rosenberg and Ken Perlin. The unmousepad: an interpolating multi-touch force-sensing input pad. *ACM Transactions on Graphics (TOG)*, Vol. 28, No. 3, p. 65, 2009.
- [18] Jefferson Y Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pp. 115–118. ACM, 2005.
- [19] Chris Harrison, Hrvoje Benko, and Andrew D Wilson. Omnitouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 441–450. ACM, 2011.

- [20] Makoto Ono, Buntarou Shizuki, and Jiro Tanaka. Touch & activate: adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pp. 31–40. ACM, 2013.
- [21] Kenneth R Foster and Henry C Lukaski. Whole-body impedance—what does it measure? *The American journal of clinical nutrition*, Vol. 64, No. 3, pp. 388S–396S, 1996.
- [22] Ken Hinckley. Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pp. 149–158, New York, NY, USA, 2003. ACM.
- [23] Dennis Schneider, Julian Seifert, and Enrico Rukzio. Mobies: Extending mobile interfaces using external screens. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pp. 59:1–59:2, New York, NY, USA, 2012. ACM.
- [24] Jun Rekimoto. Pick-and-drop: A direct manipulation technique for multiple computer environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pp. 31–39, New York, NY, USA, 1997. ACM.
- [25] Kaori Ikematsu and Itiro Siio. Memory stones: An intuitive copy-and-paste method between multi-touch computers. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pp. 1287–1292, New York, NY, USA, 2013. ACM.
- [26] Touche for Arduino: Advanced touch sensing. <http://www.instructables.com/id/Touche-for-Arduino-Advanced-touch-sensing/>.
- [27] 金属の固有抵抗と温度係数. <http://ebw.eng-book.com/pdfs/e7fceb53a50b9c1a89c75402059cd6b8.pdf>.