

公立はこだて未来大学 2014 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2014 System Information Science Practice
Group Report

プロジェクト名

屋内移動用パーソナルモビリティの活用

Project Name

Make Use of Personal Mobility for Indoor Movement

グループ名

ソフトウェア班/ハードウェア班/マイコン班

Group Name

Software Group/Hardware Group/Micro Computer Group

プロジェクト番号/**Project No.**

06-A/B/C

プロジェクトリーダー/**Project Leader**

1012139 西村祐輝 Yuki Nishimura

グループリーダー/**Group Leader**

1012127 金井貴浩 / 1012055 橋本隆義 / 1012084 山本一希
Takahiro Kanai/Takayoshi Hashimoto/Kazuki Yamamoto

グループメンバ/**Group Member**

1012002 稲葉祐太 Yuta Inaba
1012030 鈴木絢子 Ayako Suzuki
1012034 中田智子 Tomoko Nakata
1012110 菊地亜美 Ami Kikuchi
1012113 小林千紘 Chihiro Kobayashi
1012123 吉田 豊 Yutaka Yoshida
1012134 高村芳明 Yoshiaki Takamura
1012167 水澤吉博 Yoshihiro Mizusawa
1012177 芹澤雪花 Yukiha Serizawa

指導教員

三上貞芳 高橋信行 鈴木昭二

Advisor

Sadayoshi Mikami Nobuyuki Takahashi Sho'ji Suzuki

提出日

2015 年 1 月 14 日

Date of Submission

January 14, 2015

概要

本プロジェクトでは Selfi というパーソナルモビリティを利用して、屋内移動のサービスを構築する。本プロジェクトでは、パーソナルモビリティとは Segway のような立ち乗り型のパーソナルモビリティを指す。立ち乗り型パーソナルモビリティとは立ち乗りの電動二輪車や街中での利用を想定した 1, 2 人乗りの小型電動車の総称であり、小回りがきくため屋内での移動に便利である。しかし、個人で所有することは高価であることや法律によって公道で走ることができないなどの問題があるため難しく、ユーザが気軽に利用できないというのが現状があるため、パーソナルモビリティは普及していない。そのため、個人で所有する必要のないレンタルサービスが適していると考えられる。よって、本プロジェクトの目的は不特定多数の人にパーソナルモビリティを使ってもらえるようなレンタルサービスを構築し、パーソナルモビリティを普及させることである。しかし、この目標は 1 年間のプロジェクト活動で達成するのは難しい。よって、今年度の目標はその目的の第一段階として、公立はこだて未来大学の関係者向けに Selfi というパーソナルモビリティのレンタルサービスを構築することとした。そのレンタルサービスを構築するための課題を、ソフトウェア、ハードウェア、PR 活動の 3 つに分けた。ソフトウェアではアプリ、サーバ、マイコンによって、貸出申請など、サービスを行う上で必要な機能を構築する。アプリは Android アプリを使用する。アプリとサーバでログイン情報の送受信を行い、サービスのために必要なマイコンで計測した値などをアプリに送信している。ハードウェアでは、安全装置を製作するなど、実際に貸し出す Selfi の安全面の確保を行う。PR 活動では、PV を作成するなど、Selfi のレンタルサービスを行っていることを公立はこだて未来大学関係者向けに宣伝する。その結果、レンタルサービスを行うことができる最低限のシステムができたと判断したため、サービスの試験運用を行った。

キーワード Selfi, レンタルサービス, ソフトウェア, ハードウェア, PR 活動

(※文責: 山本一希)

Abstract

We propose an indoor movement support service using personal mobility. In this project, we use a personal mobility classified as standing type like Segway. This type is standing scooter for one person in city. And this type is useful to move indoor because this type have a small turning circle. Personal mobility have a small turning circle, so it is convenient for indoor movement. However, owning personal mobility personally is difficult and users could not use it readily because it is expensive and it is illegal to use on public road in Japan. So personal mobility is not spread. Therefore we think rental service which doesn't need owning personal mobility is appropriate. So, the purpose of this project is spread personal mobility by making a rental service for general public. But, it is difficult that we achieve this purpose for 1 year. We divided the problems to make the service into three parts that software, hardware and PR campaign. In software part, necessary functions implemented for the service, for example receiving a request for rental, by application, server and microcomputer. We developed an Android application. Server communicate the information of login. Microcomputer transmit information like a figure it measured to application. In hardware part, security of Selfi and safety of users of the service were improved by making a safety device and so on. In PR campaign part, advertisement of the rental service to people related to Future University Hakodate was achieved by making PV and so on. As a result, we could make minimum essential system for the service.

Keyword Selfi, Rental service, Software, Hardware, PR campaign

(※文責: 山本一希)

目次

第 1 章	はじめに	1
1.1	背景	1
1.2	該当分野の現状と従来例	2
1.2.1	パーソナルモビリティの使用事例	2
1.2.2	昨年度の成果	3
1.3	本プロジェクトの目的	3
1.4	今年度の目標	3
第 2 章	課題分析	5
2.1	課題分析	5
2.2	サービスの課題	6
2.3	実現するサービスの概要	6
第 3 章	実現するサービスのシステム構成	9
3.1	サービスのシステム構成	9
3.2	ソフトウェア	10
3.2.1	アプリケーション	10
3.2.2	サーバ	11
3.2.3	マイコン	11
3.3	ハードウェア	11
3.4	PR 活動	12
第 4 章	実現するサービスの課題解決手法	13
4.1	ソフトウェアの課題解決手法	13
4.1.1	アプリケーション	13
4.1.2	サーバ	14
4.1.3	マイコン	14
4.2	ハードウェアの課題解決手法	15
4.3	PR 活動の課題解決手法	15
第 5 章	実現するサービスの課題解決プロセス	17
5.1	ソフトウェアの課題解決プロセス	17
5.1.1	アプリケーション	17
5.1.2	サーバ	29
5.1.3	マイコン	34
5.2	ハードウェアの課題解決プロセス	44
5.3	PR 活動の課題解決プロセス	56
第 6 章	Selfi の試乗とサービスの試験運用	63

6.1	試乗での活動	63
6.1.1	オープンキャンパス	63
6.1.2	HAKODATE Developer Conference	64
6.2	試験運用の実施	64
6.2.1	試験運用の概要	64
6.2.2	試験運用の評価	66
第7章	まとめ	69
7.1	プロジェクトの成果	69
7.2	システムの各構成要素ごとの成果	69
7.2.1	アプリケーション	69
7.2.2	サーバ	71
7.2.3	マイコン	71
7.2.4	ハードウェア	72
7.2.5	PR 活動	72
7.3	発表の評価	72
7.3.1	中間発表会	72
7.3.2	最終発表会	73
7.4	今後の課題と展望	74
	参考文献	77

第 1 章 はじめに

1.1 背景

今日、世の中には様々な乗り物がある。例えば、自動車や自転車が使われている。自動車は距離の遠い場所に行くのに便利で歩行よりも移動時間が短く、天候を気にせず乗ることができる。しかし、自動車を運転するには自動車の種類に見合った免許が必要となり、免許は 18 歳以上でなければ取ることが出来ない。自転車は自動車ほどではないが歩行よりも移動時間が短く、小回りが利く。また自動車と違って免許を必要としないので誰でも乗ることが出来る。その他にも、パーソナルモビリティという乗り物がある。

パーソナルモビリティとは、立ち乗りの電動二輪車や街中での利用を想定した 1, 2 人乗りの小型電動車の総称である。これには Segway Inc. で販売されている Segway やトヨタの Winglet のような立ち乗り型や日産のニューモビリティコンセプトのような車両型などがある。本プロジェクトでは立ち乗り型のをパーソナルモビリティとして採用する。パーソナルモビリティは重心移動によって前進・後退ができ、直感的な操作が可能である。その中で本プロジェクトでは、図 1.1 に示す Selfi という株式会社エフ・アイ・ティで販売されている一人用の立ち乗り電動二輪車を使用している。Selfi の利点は、他のパーソナルモビリティと違い、一から組み立てることが出来るので内部構造の理解が可能のため、サービス形態に適する改造を施すことが出来ることが挙げられる。また、Selfi は免許が必要ないので誰でも運転可能であることや、最高速度が 10km/h なので歩行より速く移動出来ることも利点として挙げられる。他にも、サイズが全長 450mm, 全幅 525mm, 全高 1080mm とコンパクトであり、その場での旋回が可能であるため、小回りが効く。しかし、パーソナルモビリティは法律によって公道を走る事が出来ないという問題点がある。そのため、パーソナルモビリティを利用したいと考えているユーザが気軽に利用できないという状態になっている。したがって、パーソナルモビリティを使用するユーザが少なく普及していないのが現状である。



図 1.1 Selfi[1]

1.2 該当分野の現状と従来例

1.2.1 パーソナルモビリティの使用事例

中部国際空港，セントレアでは Segway というパーソナルモビリティを使ったセグウェイ・ガイドツアーというセントレア見学ツアーを実施している [2](図 1.3)。Segway とは図 1.2 に示したような Segway Inc. から発売されているパーソナルモビリティであり，本プロジェクトで使用する Selfi に似たパーソナルモビリティである。ツアーに行く前に 10 分ほどのレクチャーを受ける必要がある [3]。レクチャー終了後は 60 分ほどのツアーに出発し，中部国際空港のなかを見学できる。また予約等がないときは 10 分ほどの体験コースを選ぶことができる。



図 1.2 Segway[4]



図 1.3 セグウェイ・ガイドツアーの様子 [5]

1.2.2 昨年度の成果

本プロジェクトの引継ぎ元である「屋内移動支援システム開発プロジェクト」では工場での Selfi の屋内移動を想定し、開発を行った。そして、大きく分けて 3 つの成果を上げた。搭乗支援ステップ、自立スタンド等の移動機能にかかわるハードウェアの拡張、Arduino を用いた速度やバッテリー残量の取得、地図アプリケーションの作成の 3 つである。そして課題として残り走行可能時間の表示、製作した部品の耐久性向上、地図アプリケーションの Selfi 本体の位置表示が挙げられていた。

(※文責: 芹澤雪花)

1.3 本プロジェクトの目的

ユーザがパーソナルモビリティを利用したいと思う状況はいくつも考えられる。例えば疲れている時にちょっとした距離を移動する時や急いでいる時、荷物を運んでいる時、乗って遊びたいと思った時などでもいい。しかし、先に述べたように今現在パーソナルモビリティは公道が走れないことや高価なこともあってあまり普及していない。そこで、本プロジェクトでは不特定多数の人にパーソナルモビリティを使ってもらえるようなレンタルサービスを構築し、パーソナルモビリティを普及させることを目的とした。

(※文責: 稲葉祐太)

1.4 今年度の目標

今年度は、本プロジェクトの目的の第一段階として、公立はこだて未来大学におけるレンタルサービス実現を目指すこととした。本大学は、バリアフリー化が行われており、目標となる Selfi のサービス運用において支障が少ない。また、自分たちに身近な環境であるため問題点を見つけやすいと判断したため、本大学での運用を決定した。

(※文責: 小林千紘)

第 2 章 課題分析

2.1 課題分析

今年度の課題設定

はじめに、昨年度作成された Selfi の試乗をプロジェクト内で行った。Selfi に試乗し、気付いたことや特徴をまとめ、Selfi を使ったサービスについての話し合いを行った (図 2.1)。話し合いでは、Selfi は 1 台の値段が高いため個人所有率が低い点、初めて乗る人が多く、乗ってみると楽しいという点、歩行よりも速度が速く、屋内での移動に適しているという点が挙げられた。これらを踏まえ、個人所有率の低いパーソナルモビリティをより多くの人に乘ってもらい、利便性や楽しさを感じてもらおうサービスの検討を行った。そこで、現在ある乗り物を使用したサービスを調べた。その結果、自転車を好きな時間に好きな場所で利用できるレンタサイクルに注目した。現在のレンタサイクルは、電子的に管理され、アプリケーションから貸し出し予約を行うことや IC カードでの支払い等が可能となっている。この事例を参考に、パーソナルモビリティを所有できない人々に好きな時間に Selfi を利用してもらえようようなレンタルサービスを構築することを課題に定めた。レンタサイクルの事例から、メールとアプリケーションを通じた利用管理を行うこと、マイコンを使用して貸し出す Selfi の情報提供を行うことを決めた。レンタルサービスを展開する場所は、実験がしやすく身近である公立はこだて未来大学を選んだ。

試乗した際に気付いた問題点

Selfi に試乗し、学内を走行するにあたっての問題点をまとめた。問題点は大きく 2 つに分けることができ、1 つはパーソナルモビリティの持つ問題点、もう一つは Selfi 自身が持つ問題点であった。パーソナルモビリティの問題点として、乗っている人の視界が狭くなりやすいこと、パーソナルモビリティに乗っている人が周りにいても認知しにくいこと、曲がり角から急に出てきた際に衝突の危険があること、学内の床に伸びたコード類を乗り越えるのは大変であること、エレベータでの移動がエレベータ利用者にとって邪魔になること、複数台の並走が難しいためすれ違う際に危険が伴うことなどが挙げられる。そして Selfi の問題点として、安定性が悪く慣れが必要であること、バッテリーの消費量が激しく駆動時間が短いなどが挙げられる。

(※文責: 芹澤雪花)



図 2.1 話し合いの様子

2.2 サービスの課題

公立はこだて未来大学でサービスを展開する際、課題となったことはレンタルの方法であった。レンタルを行うためにどのような方法があるかを話し合った結果、メールの機能を利用することになった。しかし、既存の方法が存在しないために、メールを利用したサービスの仕組みを作成する必要性があった。

次に、サービスを利用するときの安全性が課題となった。サービスの利用者には、Selfi に乗りなれていないユーザもいる。なので、初心者のユーザが安心して利用できるように、Selfi の安全性の向上が課題となった。

また、ユーザが Selfi の使い方を知らない可能性があった。なので、使い方を知らないユーザに対してのマニュアルを作成する必要性があった。そして、サービスを利用するユーザが多く存在する場合、貸出可能な Selfi の台数が少なくなってしまう、利用できないユーザが出てきてしまう可能性がある。そのため Selfi の台数を増やすことも課題であった。

最後に、サービスで貸し出す Selfi のセキュリティが課題となった。貸し出した Selfi などの盗難は未然に防ぐ必要があり、セキュリティの強化が必要となった。

(※文責: 橋本隆義)

2.3 実現するサービスの概要

本プロジェクトで実現するサービスは Selfi のレンタルサービスである。ターゲットは公立はこだて未来大学の学生・教職員とし、利用場所は公立はこだて未来大学内に限定する。運用時間は 9:00 から 18:00 までとする。貸出場所はエレクトロニクス工房とし、貸出 1 回あたりの利用時間は最大 1 時間 30 分までとした。料金は無料とする。利用方法は、メールを送ることで借りることができ、Selfi に搭載されている Android 端末のアプリケーションでログインすると乗ることができる。以上のサービスの情報を表 2.1 にまとめて示す。また、サービスの利用の流れを図 2.2 に示す。

(※文責: 水澤吉博)

表 2.1 サービス概要

ターゲット	公立はこだて未来大学関係者
利用場所	公立はこだて未来大学構内
利用時間	1 時間 30 分
料金	無料
貸出場所	3F モール エレクトロニクス工房前
運用時間	9:00 ~ 18:00
利用申請方法	メールで申請

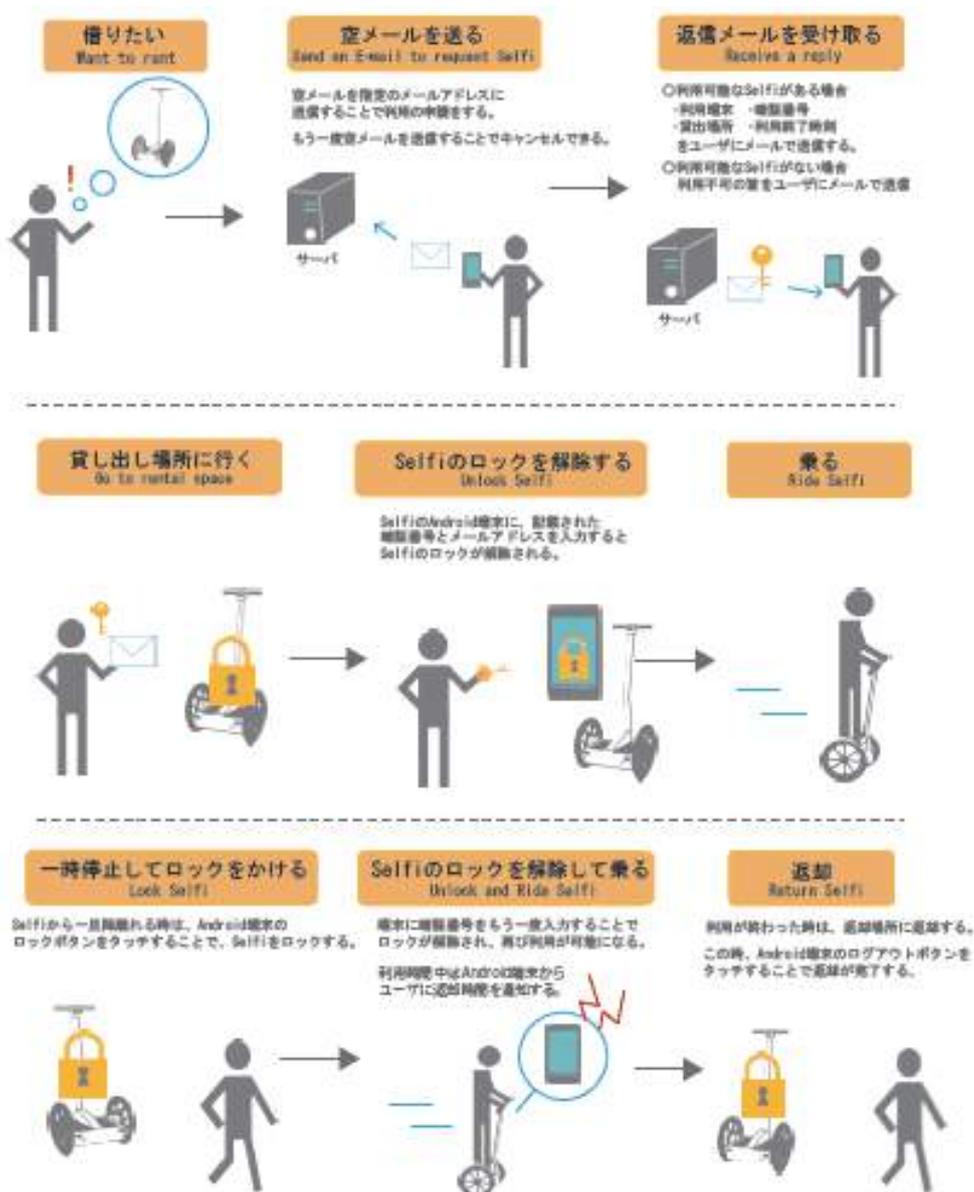


図 2.2 サービス利用の流れ

第 3 章 実現するサービスのシステム構成

3.1 サービスのシステム構成

本プロジェクトで実現するレンタルサービスのシステムの構成を図 3.1 に示す。この図の左上がサーバ、右上がアプリ、左下がユーザ、右下が Selfi 本体とそれに搭載されているマイコンを示している。以下に、このシステムの処理手順を示す。

1. ユーザがレンタルサービスを利用したい時に、指定されたアドレスにメールを送信する。
2. サーバはその空メールを受信したら現在利用可能な Selfi を検索し、利用可能な Selfi が見つかった場合にはロックを解除するために必要なパスワードを発行しユーザ側に返信する。また、利用可能な Selfi がなかった場合にはレンタルサービスを利用できない旨を記したメールを返信する。
3. ユーザ側がパスワードを記されたメールを受け取ったら、Selfi 本体に取り付けられている Android 端末にメールを送った時のアドレスをユーザ名に、送られたパスワードをそのままパスワードの欄に入力してログインすることで Selfi のロックを解除する。
4. ログインを行った後はサーバ側と Selfi 間でログイン情報をやりとりして、サーバ側で Selfi の現在の情報を管理する。
5. Selfi 利用中には、Selfi 本体に内蔵されている Arduino と通信を行うことで現在の速度、バッテリー残量の数値をアプリ側に送信し、表示することでユーザ側でそれらの情報を確認することを可能とした。

(※文責: 稲葉祐太)

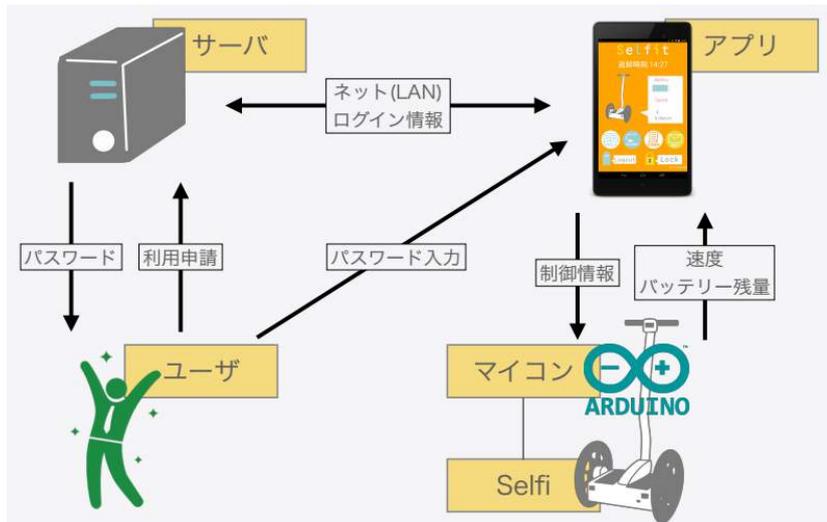


図 3.1 システム構成

3.2 ソフトウェア

ここで、3.1 節で示したシステムの各構成要素の内、アプリケーション、サーバ、マイコンを併せてソフトウェアと呼ぶこととする。ソフトウェアでは、2.2 節で述べたサービスの課題の内、主に、サービスの仕組み作り、初心者対策、セキュリティ強化などを行う。これらの課題解決のために必要な機能について、以下に述べる。

3.2.1 アプリケーション

レンタルサービスを行う上で、アプリケーションによって以下の機能を実装する。

- ユーザが Selfi を利用するための、レンタルサービスへのログイン機能。
- 初心者のユーザは Selfi の乗り方を知らないため一人で Selfi に乗ることは出来ない。そこで、初心者でも人の手を借りずに乗りこなせる対策を行う。
- ユーザがサービス利用中に、返却場所など必要な情報を知ることが出来る機能。
- ユーザが Selfi を乗車中に周囲の人々と接触しないよう、Selfi の接近を知らせる機能。

(※文責: 高村芳明)

3.2.2 サーバ

サーバによって、レンタルサービスに必要な以下の機能を実現する。

- ユーザから自動で利用の申請を受け取ることや、その申請に対する応答のためのメールの送受信機能。
- ユーザの利用申請に対して、どのように応答するのかを決定するためのデータの作成と、データから Selfi が利用可能かどうかを判断する機能。

(※文責: 吉田豊)

3.2.3 マイコン

以下の機能を、マイコンによって Selfi に取り付ける。

- Selfi のロック機能をつけることによって、Selfi の盗難を防止する。
- Selfi の速度を計測することで、ユーザに今の速度が危険であることを伝える。
- Selfi のバッテリー残量を計測することで、サーバに Selfi の貸出が利用可能かどうかを判断させる。

(※文責: 山本一希)

3.3 ハードウェア

ハードウェアでは、2.2 節で述べたサービスの課題より、Selfi 本体について以下の課題を解決する。

- Selfi の組み立て。
- 安全の確保。
- Android 端末の Selfi への取り付け。

(※文責: 中田智子)

3.4 PR 活動

PR 活動として、ユーザにレンタルサービスを認知してもらうために、以下の活動を行う。

- イメージキャラクターの製作.
- ナンバープレートの製作.
- 看板の製作.
- 宣伝動画の製作.
- チームパーカーの製作.

(※文責: 鈴木絢子)

第 4 章 実現するサービスの課題解決手法

4.1 ソフトウェアの課題解決手法

4.1.1 アプリケーション

4.1.1.1 Android アプリ

Android アプリでは、3.2.1 節で述べた機能の内、以下の機能の実装を目指し、課題解決に取り組む。

ログイン機能

ユーザの学籍番号とメールで返信されたパスワードを入力すると、サーバと通信を行い認証を行うログイン機能を作成する。認証が成功した場合、アプリケーションとマイコンを通信させることにより、Selfi のロックを解錠・施錠する。また、ユーザが Selfi の使用を終えた後は、ログアウトボタンを押すことで Selfi をロックし、サーバにログアウト情報を送信するログアウト機能を実装する。

初心者対策

マニュアルページを導入することで、利用者に Selfi の乗り方や降り方を伝える。

ユーザへの情報提示

利用規約ページを作成し、Selfi をレンタルする上でのルールを確立させる。また、Selfi の返却場所と緊急連絡先を提示するページを作成し、不測の事態に備える。

(※文責: 小林千紘)

4.1.1.2 音

3.2.1 節で述べた機能の内、周囲の人々に Selfi の接近を知らせる機能の実現のため、接近報知音を作成することによって課題解決を行う。

Selfi の接近報知手法

Selfi は静かな環境では走行音が聞こえるが、大学の昼休みや講義と講義の間などは周囲の環境が騒がしく Selfi の走行音が聞こえなくなる。その際後ろから走ってくる Selfi に気づかず、危険な状態である。よって Selfi の接近を知らせる必要がある。そこで、Selfi の走行中にアプリケーションから報知音を鳴らすことで周囲の人々に Selfi の接近を伝えることとする。

接近報知音の作成

接近報知音は周りの環境に合わせた音が必要である。まず周囲の音をはかり、その環境音に合わせた接近報知音を検討する。その作成した接近音をマイコンからくるスピードに対応させて音を流すようにする。

(※文責: 金井貴浩)

4.1.2 サーバ

サーバでは、3.2.2 節で述べた機能の実現のため、以下の方法で課題解決に取り組む。

メールの送受信

メールサーバを設置し、ユーザからの利用申請を受け取った際にその申請に対する返信を行うプログラムを作成する。

Selfi の利用可能状態の判断

データベースサーバで Selfi の状態を管理し、かつメール受信時にデータベースを参照し Selfi の状態を判断するプログラムを作成する。

(※文責: 水澤吉博)

4.1.3 マイコン

マイコンでは、3.2.3 節で述べた機能の実現のために、以下の方法で課題解決に取り組む。

Selfi のロック機能

アプリからログイン情報を取得し、その情報をもとに Selfi の制御をマイコンによって切り替える。

速度計測

タイヤの回転数を計測することで一定時間に進んだ距離を求め、そこから平均の速度を割り出す。

バッテリー残量の計測

バッテリーの電圧を計測し、バッテリーの最大電圧を 100%、Selfi が走行可能な最低限の電圧を 0% として百分率で表す。

(※文責: 西村祐輝)

4.2 ハードウェアの課題解決手法

ハードウェアでは、3.3 節で述べた課題の解決のために、以下の方法で課題解決に取り組む。

Selfi の組み立て

株式会社エフ・アイ・ティーより Selfi を発注し、それを組み立てる。

安全の確保

サービスを利用するユーザに安心して Selfi に乗ってもらうために、安全装置を作成する。

Android 端末の Selfi への取り付け

サービスを展開していく上で、Android 端末が使いやすいことが望ましい。そのため、Selfi に Android 端末を入れることが可能なケースを作成し、取り付ける。

(※文責: 橋本隆義)

4.3 PR 活動の課題解決手法

PR 活動として、3.4 節で述べた内容について以下の方法で課題解決に取り組む。

PV 動画

iMovie という映像編集ソフトを用いて PV 動画を製作。

イメージキャラクタ

今回のサービスを展開していく中で、サービスの認知度の促進を図るために、イラストレータを用いて製作。

ナンバープレート

本プロジェクトは Selfi を 3 台所有しているため、サービスとして展開していく際に識別が必要なため、アクリル板と印刷用シールを用いてナンバープレートを製作。

看板

11 月 26 日から 11 月 28 日の 3 日間の試験運用の場所をわかりやすくするために、MDF 材とアクリル板を用いて製作。

チームパーカー

本プロジェクトメンバの団結力を高めるため、また、サービスの宣伝としてパーカーを製作。

(※文責: 鈴木絢子)

第 5 章 実現するサービスの課題解決プロセス

5.1 ソフトウェアの課題解決プロセス

5.1.1 アプリケーション

5.1.1.1 Android アプリ

以下に、4.1.1.1 節で述べた課題解決のために、実際に行った Android アプリ開発のプロセスを示す。

- アプリの構成の考案.
- ログイン機能の作成.
- 初心者対策のためのマニュアル作成.
- ユーザへの情報提示を行うページの作成.
- UI デザインの設計.

これらの各項目について、以下に詳細を述べる。

アプリの構成

Selfi のレンタルするにあたり、ユーザの管理と必要な情報提供をするためのアプリケーションを開発した。このアプリケーションは、ユーザの管理を行うサーバと Selfi の制御を行うマイコンと通信を行い、ログイン機能を設けてユーザの利用状況を管理し、ログイン時に Selfi の電池残量や速度、乗り方や返却方法などの情報を提供し、レンタルサービスを使いやすくする機能を設けた。アプリケーションの名前は、Selfi に合った、Selfi の学校との調和、共存を意味し” Selfi” と” fit” を合わせて” Selfit” と名付けた。

以上の機能を持つアプリケーションの作成のため、始めに図 5.1 のようなデザインスケッチを行った。

(※文責: 芹澤雪花)



図 5.1 アプリのデザインスケッチ

ログイン機能

ログイン機能の概要 ユーザが Selfi のサービスを利用するために Android アプリとサーバをローカル環境の無線 LAN で通信を行っている。サーバから情報を取得するときは文を変数展開したものをサーバに送信することで必要な情報がサーバから取得出来るようになっている。ユーザが Android アプリにログインするとき、図 5.2 にある ID とパスワードの空欄にレンタル申請の返信の中に記載されているパスワードとユーザ自身の@以前のメールアドレス (以下、ID) を入力してログインボタンを押す事でサーバと通信が行われ必要な情報を取得している。取得した ID、パスワードとユーザが入力した ID、パスワードが合致した場合、ユーザはログインすることが可能となっている。なお、以下の文は Android アプリの SqlAdapter クラスのプログラムから抜粋した文である。SqlAdapter クラスはサーバと通信のやり取りをするプログラムである。サーバから取得する情報は SelfiID、ユーザ ID、パスワード、レンタル No の 4 つである。

以下の文を変数展開したものをサーバに送信することでログインに必要な SelfiID を取得するものである。SelfiID はログイン時、ログアウト時、充電完了時にサーバから取得する必要がある毎回通信を行っている。

```
"select Selfi_ID from Selfi where Macaddress = ' " + mac + " " "
```

ログイン時の処理 ここでは、ユーザがログイン時に必要な情報のやり取りについて記述する。取得したパスワード、ID とユーザが入力したパスワード、ID が合致していればログインを可能とした。

以下の文を変数展開したものをサーバに送信することでパスワード、ID と貸し出し可能な Selfi のレンタル No の 3 つの情報を取得することが出来る。

```
"select Password, Mailaddress, Rental_No from Rental " + "left outer join Selfi on Rental.Selfi_ID = Selfi.Selfi_ID left outer join User on " + "Rental.User_ID = User.User_ID where Rental.Selfi_ID ='" + self_id + "' and Status = 'Waiting' and " + "Rental.Rental_No
```



図 5.2 ログイン画面

```
= (select max(cast(Rental_No as signed)) from Rental where Selfi_ID = ''' + self_id + ''')
```

ユーザがログイン可能となった場合、貸し出し時刻の更新と Selfi の状態を Waiting の状態から Rental の状態に更新する必要があるためその情報をサーバに送信する。それは Selfi の状態を Rental 状態にすることで他のユーザから申請があった時、ユーザが使用している Selfi は貸出中になっているので、他のユーザが貸し出し出来ないようにするためである。

以下の文は変数展開したものをサーバに送信することで取得したレンタル No の貸し出し時刻を更新する文である。文中にある「?」はそれぞれの値を入れるものであるため、レンタル時刻ならば「?」に現在の時刻が値として入る。これ以降の「?」も同様である。

```
"update Rental set Rental_Time = ? where Rental_No = ?"
```

以下は変数展開したものをサーバに送信することで取得した SelfiID の Selfi の状態を更新する文である。

```
"update Selfi set Status = 'Rental' where Selfi_ID = ? and Status = 'Waiting'"
```

ログアウト時の処理 ユーザが Selfi のサービスを終了するためにログアウトボタンを押すときに行う処理である。ログアウトの時は返却時刻と Selfi の状態を更新する。Selfi の状態が Available の状態の時にユーザから申請メールが送られてくると Selfi の状態がサーバ側で Available の状態から Rental の状態に変更され貸し出し可能となる。Selfi が充電中の時はユーザがレンタルの申請メールを送ってきても使用できない状態である。

以下の文を変数展開したものをサーバに送信することで取得したレンタル No の返却時刻を更新する文である。

```
"update Rental set Return_Time = ? where Rental_No = ?"
```

以下の文を変数展開したものをサーバに送信することで取得した SelfiID の Selfi の状態を Rental の状態から Available の状態に更新する文である。

```
"update Selfi set Status = 'Available' where Selfi_ID = ? and Status = 'Rental'"
```

本来はユーザが Selfi を利用し終えログアウトすると Available の状態になるのではなく

Make Use of Personal Mobility for Indoor Movement

Charging の状態になるのだが Selfi の電源がオンになっているときは充電ができないという不具合があり今回は Charging の状態にはしなかった。したがって今年度では充電部分は実装しなかった。

Selfi の充電完了時の処理 以下の文を変数展開したものをサーバに送信することで Selfi の充電が 90% を超えた時サーバに Selfi の状態を Charging の状態から Available の状態に更新する文である。

```
"update Selfi set Status = 'Available' where Selfi_ID = ? and Status = 'Charging'"
```

Selfi のロック機能 ユーザが Selfi を利用中、一時的に Selfi から離れるときに使用する機能であり Home 画面にあるロックボタンを押すと図 5.3 に移行して Selfi の制御が入らない状態になる。ロックを解除したい場合はログイン時に使用した ID とパスワードを入力するとロックが解除され、再び Selfi に制御が入り、Selfi を利用することが出来る。



図 5.3 ロック画面

Selfi の制御 Selfi の制御を入れることで Selfi に乗ることが出来る。その制御の切り替えを Arduino が行っている。Android アプリは Bluetooth Low Energy (以下, BLE) を使用し Arduino と無線通信を行って制御の切り替えを操作する。ユーザがログイン状態になると 1 を、ログインしていない状態の時は 0 を Arduino に送信している。

(※文責: 高村芳明)

初心者対策

Selfi の乗り方や降り方がわからない初心者のために、マニュアルページを作成した(図 5.4)。マニュアルページでは、乗り方・運用方法・降り方と用途によってページを分けることで、見たい項目をすぐ参照できるよう工夫した。

マニュアルページは、ホームにあるマニュアルボタンを押すことで開くことができる。そのページより、乗り方・運用方法・降り方の各項目に遷移する仕組みとなっている。各ページには、マニュアルトップへのボタンとホームへ戻るボタンを設置した。

乗り方ページでは Selfi への乗り方を、運用方法ページでは Selfi で走行する方法、降り方ページでは Selfi からの降り方を、それぞれ写真や注意事項を添えながら説明した。実際に掲載した内容の概略は以下の通りである。

乗り方

Selfi の制御が入るまでハンドルを前後させる。

注意事項：スピードを出し過ぎたり、急ブレーキをかけないこと。

操作方法

重心を前に傾けると進み、後ろに傾けると後進する。ハンドルを傾けると方向転換が行える。

降り方

片足ずつゆっくりと Selfi から降りる。

一時的に Selfi を降りる場合：タッチパネルのロックボタンをタッチすることで Selfi をロックできる。

Selfi を返却する場合：タッチパネルのログアウトボタンをタッチすることで、返却完了となる。

各ページへの遷移は Java の R.id と xml の Button を使って行い、マニュアルに使用している画像は xml で PNG ファイルを参照することで記載した。



図 5.4 マニュアルページ

(※文責: 小林千紘)

ユーザへの情報提示

ユーザへの情報提示を行うページでは、それぞれ xml の TextView を使用し、小見出しをつけたり文字の背景色を白にすることで、文字が見易いようにした。

また、画面遷移は Java の R.id を使用し、xml のボタンとリンクすることで実現した。

利用規約ページの作成 Selfi のレンタルサービスを運用する上で、利用者に同意してもらうための利用規約ページを作成した (図 5.5)。利用規約ページは、ホームにある利用規約ボタンを押すことで遷移することができる。利用規約ページを作成するにあたって、利用規約の内容を話し合い、Selfi をレンタルする際に発生しうる故障や盗難の対処方法に重点を置いて記載した。記載した内容の概略は以下の通りである。

1. サービスの利用申込方法・利用料金・利用時間・利用条件.
2. Selfi の盗難・紛失について.
3. Selfi 利用中の怪我・事故について.
4. 禁止事項・違反時の対応について.
5. 緊急時の連絡方法について.

利用規約の文章は xml に組み込み、ScrollView を使用することで、画面をスクロールさせながら規約を読むことができるように工夫した。また、ホームへ遷移できるボタンも設置した。

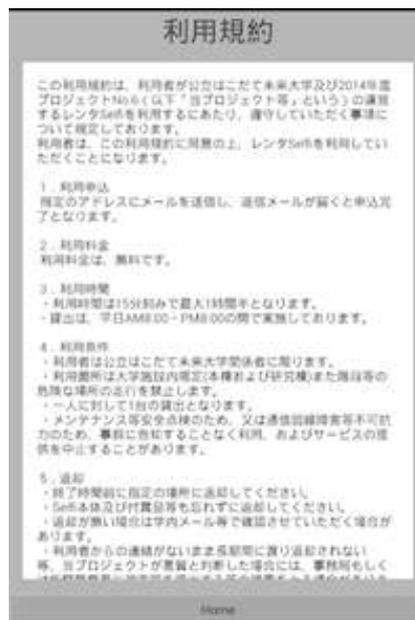


図 5.5 利用規約ページ

返却場所ページの作成 Selfi を返却する場所の地図を表示したページを作成した (図 5.6)。返却場所ページは、ホームの返却場所ボタンを押すことで遷移できる。

本学 3 階の構内図を参考に、Illustrator を使用して 3 階地図を作成し、返却場所ページに記載した。このページは、利用者が Selfi を返却する際に見られるページだと想定されるため、返却方法の概要も掲載した。使用した地図の画像は、xml で画像指定を行い表示し、ホームへ遷移できるボタンも設置した。



図 5.6 返却場所ページ

緊急連絡先ページの作成 Selfi を紛失した場合・盗難に遭った場合・利用中に怪我や事故が発生した場合などに、プロジェクトメンバへ連絡するためのメールアドレスを記載した緊急連絡先ページを作成した (図 5.7)。緊急連絡先ページは、ホームの緊急連絡先ボタンを押すことで遷移できる。

緊急連絡先のメールアドレスは、フリーメール (Gmail) を取得し掲載した。このメールアドレスに連絡があった場合、プロジェクトメンバに通知され、連絡者の状態を知ることができる。

緊急連絡先ページでは、文字が見易いように TextView と ScrollView を使用し、文字の背景色を白に設定し、ホームへ遷移できるボタンも設置した。

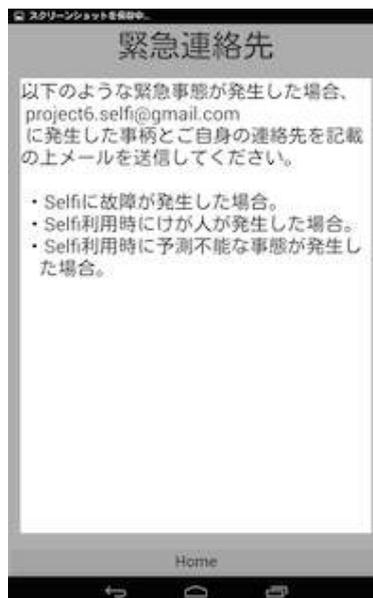


図 5.7 緊急連絡先ページ

(※文責: 小林千紘)

UI デザイン

UI デザインのテーマとして、Selfi の利用を楽しくするために明るい色合いを用いてフラットデザインで見やすいデザインを心掛けた。色は、オレンジを中心とした。メニュー画面には、上部に大きくロゴ、真ん中に Selfi のイラストと Selfi の状況を示すアイコンを配置し、下部に利用規約、利用方法、返却場所、連絡先の 4 つのメニューアイコンを表示している。以上のデザインを施したメニュー画面を図 5.8 に示す。



図 5.8 メニュー画面

(※文責: 芹澤雪花)

5.1.1.2 音

以下に、4.1.1.2 節で述べた課題解決のために、実際に行った Selfi の接近報知音製作のプロセスを示す。

- 環境音の調査.
- 接近報知音のパラメータの検討.
- 接近報知音に相応しい周波数を求める実験.

これらの各項目について、以下に詳細を述べる。

環境音の調査

接近報知音とは自身の存在を周囲に伝え、安全に通行することを目的として作る音のことである。現在では電気自動車やハイブリット車などに使用されている [6]。今回のプロジェクトでは Selfi を安全に走行するため周囲の人々が Selfi の接近に気づけるような接近報知音の検討をする。

接近報知音に求められる音は不快ではない音、対象となるものから接近報知音が出ているとわかる音が必要な条件である [7]。接近報知音を検討するにはまず周りの環境音を調べる必要がある。今回のプロジェクトのレンタルサービスでは主に公立はこだて未来大学の3階モールを主に使用するので接近放置音を検討するのに必要な環境音は3階モールの環境音を使用することにした。3階モールには朝や講義中などのとても静かな時間と昼休みや講義と講義の間の休み時間など騒がしい時間など時間帯によって環境音が大きく変わる。まず複数の時間を調べ、どの時間帯、もしくは平均をとって参考にする環境音にしなければならない。しかし3階モールにおける環境音を収録する際にマイクをどの方向に向けるべきかわからず、また方向によって収録する音が変わる可能性があるので収録する方向を決めるため実験を行った。

目的 公立はこだて未来大学の3階モールにおける環境音を収録する際、方向によって環境音の大きさに差異はあるかどうかを調査する。

方法 行った実験の方法について、以下に詳細を述べる。

場所

公立はこだて未来大学の3階モール。

実験器具

精密騒音計 (LA-5560, 小野測器社) とそれを固定する三脚を使用した。

手続き

3階の工房と365教室の中間に三脚をたてその上に騒音計を設置した。その際騒音計は水平にし、先端を地面から140cmの高さに設定した(図5.9)。騒音計の設定はLAeq, レンジは20 - 80dBとし5分間の測定時間として設定した。工房を0°とし研究棟の方向を90°, 365教室を180°, 玄関を270°とする。0°, 90°, 180°, 270°の四方向に向け5分の計測が終わるたびに方向を90°ごとずらしていった。また4限が2回, 5限が1回の計三回測定した。



図 5.9 設置方法

結果 三回の測定結果は以下の表 5.1 のようになった。

表 5.1 実験 1

0°	90°	180°	270°
47.9dB	51.8dB	47.7dB	49.0dB
52.9dB	51.0dB	50.1dB	50.0dB
55.9dB	52.0dB	52.0dB	53.0dB

今回の測定では三回とも 51dB を超えているのは 90° 方向しかなく、ほかのものはすべて 40 台の値を持っている。これらのデータを分散分析検定をしてみたところ、郡内自由度が 3、群間自由度が 8 となり F 値が 4.46 となった。よってこれら四つの方向に関して音を収録するさい特に差異がないことがわかった ($F(3, 8) = 4.46, p > .05$)。

まとめ 以上の実験により、3 階モールにおける環境音を収録する際、方向は特に加味しなくてもいいことが判明した。

次に 3 階モールにおける環境音の収録をした。また時間帯によって環境音が違うかどうか調べ、なおかつさまざまな時間帯の音量の計測として朝と昼休みと授業中として 4 限の 3 階を収録した。マイクと騒音計を使い同時に環境音と音量を同時に取ろうとした。だがうまく環境音を収録できなかった。理由としては今回使用したマイクが AT-X11 (audio-technica 社) というマイクを使用したのだがこれはボーカル用マイクであった。なので環境音の収録などの騒音の収録には向いていない。しかし、それぞれの時間ごとの環境音の音量がとれたのでこれを参考にする。環境音が一番でかいと思われる昼休みの環境音が 56dB だったのでこの音量のなかでも聞こえる接近報知音をつくる必要がある。

(※文責: 金井貴浩)

接近報知音のパラメータの検討

接近報知音は使われる環境での環境音の中でも聞こえるようにしなければならない。そこで環境音を参考に接近報知音のパラメータを設定するのだが、今回はうまく環境音の収録がうまくいかなかった。しかし音量については精密騒音計を用いることで測定できた。なのでこの測定した環境騒音音量を使用しパラメータの決定していく。接近報知音は Selfi 本体が歩行者等に接近する際、事前にその存在を知らせ、注意喚起などを目的として作る音なので Selfi に対してどのような距離で聞こえ気づけたら安全に回避できるかを考える必要がある。よって簡単な実験を行った。Selfi が一番危険な後ろから近づいてきたら危険を感じるのは何 m からなのか、を調べる。3 階モールで行い後ろから Selfi に近づいてもらい振り返った時に危険を感じた距離を測定した。この結果と過去のプロジェクトの動画を参考にし Selfi が後ろから近づいてきたとき危険を感じるのは 5m とした。この結果によって 5m 先にいる人に聞こえて Selfi に気づけるのを目的とした接近報知音を作る。

音の要素は主に 3 つある。よって、以下の 3 つの項目のそれぞれについてパラメータを決めていかねばならない。

音量 (音圧)

環境音が一番うるさいときに聞こえればそれ以下の音量でも聞こえるはずであると考え、想定する環境を昼休みの 3 階モールとした。失敗した音の収録時と同時に昼休みの環境音の音量の測定を行っていたので音量の大きさはわかる。よって 56dB を接近報知音の参考にするパラメータにする。接近報知音に求められる音量というのはその環境の環境音より 10dB 程度高いものが一番適切な音量となる [7]。

音の高さ

今回の環境音の収録では 3 階モールの環境音が収録できなかった。よって今回は印象評価実験を行い、不快でない音かつ音の検出率がいいものを検討する。Scilab を使い 261Hz から 493Hz の「ビー」という単調な音を作成し、アンケートを答えてもらう、という形でパラメータを決める。

音色

接近報知音は対象となるものが近づいてくるとわかるものがもっとも効果的な接近報知音である [7]。よって Selfi のモーター駆動音を加工し接近報知音にしようとしたのだがノイズがひどく、音域を変更しようとする音が割れてしまった。今回は全体的に収録に難があるので収録音はあらかじめ音を Scilab 等で生成したものをを使うことにする。

(※文責: 金井貴浩)

周波数決定のための実験

実験の概要 検討した音について報知音としてふさわしい周波数を求めるための実験を行うことが良いと考えた。検討した音が実際にサービスが行われる状況下でどのように感じられるかを検証するため、印象実験を行った。実験の詳細については以下の通りである。

被験者は 20 代前半の男子大学生、女子大学生 7 名で実験を行った。実験には、パーソナルコンピュータ (Dynabook, R731/E26ER) 2 台、スピーカー、騒音計 (ONOSEKKI, LA-5560)、三脚を用いた。パーソナルコンピュータは、環境音を想定した音源を再生するために用いた。スピー

カーは検討した音の大きさを Selfi の報知音として想定した大きさに調節するために用いた。騒音計は使用する教室の音圧レベルを測定するために用いた。実験は 2014 年 11 月 7 日の 14 時 50 分から 18 時 00 分の間で行われた。場所は公立はこだて未来大学の 5 階, 584 教室で行った。実験時の教室の音圧レベルは 40.7dB であった。

上記で作成した音を用い実験を行った。それぞれ 261Hz, 293Hz, 329Hz, 349Hz, 392Hz, 440Hz, 493Hz の周波数であり、ここではそれぞれの音を A1, A2, A3, A4, A5, A6, A7 とした。これらの音を示す記号と周波数との対応関係を表 5.2 に示す。

表 5.2 音を示す記号と周波数の対応

記号	A1	A2	A3	A4	A5	A6	A7
周波数	261Hz	293Hz	329Hz	349Hz	392Hz	440Hz	493Hz

さらに環境音として 3 人の女性が本を朗読している音声ファイルを準備した。報知音、環境音の音圧レベルを 56dB に設定した。

スピーカーと人との距離は、歩行者が Selfi に気付くことができ、さらに危険を回避できると予測される 5m に設定した。

実験は 2 人ずつ行った。最初に A1 から A7 の 7 種類の音のパターンを聴いた後、4 検法で質問用紙に回答することを指示した。被験者には合図のあとに音を流すことを説明した。次にあらかじめ決めておいた順番で 7 つの音源をランダムに再生し、被験者に 1 回ずつ聴かせた。これにより 7 つの音に対する印象実験を行った。また質問用紙に回答し終えたことを確認し、次の音へと移行した。質問内容はそれぞれの音に対して聞こえ方について、不快感について、音に気付くかどうかとした。分析には要因参加者内で分散分析を行ったあと、対応のある t 検定を行った。

(※文責: 菊地亜美)

結果 まず質問 1 の音が聞こえるかどうかということに対して、郡内自由度を 6, 群間自由度を 21 とし、7 つの音に対して要因参加者内での分散分析を行ったところ、F 値は 13.79 となり有意な差が見られた ($F(6, 15) = 13.79, p < .01$)。

これにより A1, A2, A3, A4, A7 が報知音として有効であるということが分かった。報知音として利用するには聞こえるという条件は外すことができないため、他の音に比べて聞こえるという結果が出なかった A5, A6 の結果は不快感、気付くかどうかの質問の t 検定から除外した。

次に質問 2 の不快であるかどうかということに対して、郡内自由度を 4, 群間自由度を 10 とし、A1, A2, A3, A4, A7 の 5 つの音に対して要因参加者内で分散分析を行ったところ、F 値は 1.5 となり有意な差は見られなかった ($F(4, 6) = 1.5, p > .01$)。

さらに質問 3 の気付くかどうかということに対して、郡内自由度を 4, 群間自由度を 10 とし、A1, A2, A3, A4, A7 の 5 つの音に対して要因参加者内で分散分析を行ったところ、F 値は 1.39 となりこれも有意な差は見られなかった ($F(4, 6) = 1.39, p > .01$)。

印象実験により A1, A2, A3, A4, A7 の音が聞こえる音ということが分かった。しかし、接近音の他の要素については有意な差は見られなかったため、実験結果よりより高い評価を得られたものを報知音として実装することがいいのではないかと考えた。そのため不快感、気付くかどうかという項目において最も高い評価を得た A3(329Hz) を報知音として実装することに決定した。

(※文責: 菊地亜美)

5.1.2 サーバ

以下に、4.1.2 節で述べた課題解決のために、実際に行ったサーバ設計のプロセスを示す。

- 使用機材の決定.
- ネットワーク環境の構築.
- メールを送受信機能の設計.
- データベースの設計.
- サーバで動作するプログラムの作成.

これらの各項目について、以下に詳細を述べる。

使用機材

本プロジェクトで使用した機材を以下に示す。

- サーバ用ノート PC
- 有線 LAN ルータ
- 無線 LAN ルータ
- 接続用 LAN ケーブル

(※文責: 吉田豊)

ネットワーク構成

インターネットを介したレンタルサービスの通信において要求されるものとして、以下のものが挙げられる。

- グローバル IP
- ドメイン名
- DNS サーバ

今回は、全ての通信をインターネットから切り離された LAN 内で行うこととした。具体的には、メールの送受信、Android 端末の MySQL サーバへの接続を同一ネットワーク内で行うこととした。

サーバ、Android 端末間の通信を LAN 内で行うために、必要な機能として、名前解決のための DNS サーバ機能、IP 割り当てのための DHCP 機能があるが、これらの機能を有する有線 LAN ルータを使用することで、通信を可能にした。

有線 LAN ルータの IP アドレスを 192.168.1.1、サーバの IP アドレスを 192.168.1.2 とし、192.168.1.4~192.168.1.254 を DHCP による割り当てとした。

また、Android 端末は WiFi によって接続する必要があったため、無線 LAN ルータのアクセスポイント機能を用いて、WiFi による接続を可能にした。

サーバのポートについては、通信を行う際に必要となる箇所を開放し、必要のない部分は通さないように設定した。表 5.3 に設定を示す。

表 5.3 ポートの設定

サービス名	ポート番号
SSH	22
MySQL	3306
SMTP	25
POP3	110
IMAP	143
IMAP over SSL	993

(※文責: 吉田豊)

メールの送受信

サービス利用の申請メールの受信とそれに対する返信を行うために、以下の課題解決を行った。

- メール送信のためにメール転送エージェント (MTA) として Postfix を用いた。MTA とは、ユーザが送信したメールの配送、メールサーバにアカウントを持つユーザへのメールの受信および保管を行うサーバソフトウェアである。
- Postfix の設定をし、メールサーバ自身のホスト名を mail.project6.localdomain、ドメイン名を project6.localdomain と設定した。
- メール受信のために Dovecot を用いた。
- メール送受信を行うために、TCP のポート番号 25 番、143 番を開放した。
- メール盗聴を防ぐために SSL の設定をし、メールを暗号化した。
- ユーザに送り返すメールの文面を考えた。初めに想定していた予約完了のメールに加え、予約キャンセルのメール、貸出中で貸出できない旨のメールを送信することとした。

(※文責: 水澤吉博)

データベース

データベースを作成・管理するために MySQL を用いた。システムに必要なデータであるレンタル情報、Selfi の状態、ユーザ情報をそれぞれ管理する Rental テーブル、Selfi テーブル、User テーブルを作成した。以下に各テーブルの構成要素を示す。

Rental テーブル

レンタル番号を表す Rental_No, ユーザごとの ID を表す User_ID, Selfi_ID, レンタルごとのログインに必要なパスワードを表す Password, メールを受け取った時刻を表す Receive_Time, ユーザがログインした時刻を表す Rental_Time, 返却時刻を表す Return_Time を入れる。主キーは Rental_No で, User_ID, Selfi_ID はそれぞれ User テーブル, Selfi テーブルへの外部キーである。

Selfi テーブル

Selfi_ID, Selfi の状態を表す Status, Android 端末の Mac アドレスを表す Macaddress を入れる。主キーは Selfi_ID である。

User テーブル

User_ID, ユーザのメールアドレスを表す Mailaddress を入れる。主キーは User_ID である。

また, Android 端末とデータベースの通信を SSL で暗号化するために, プライベート証明書を作成したが, うまく認証できなかった。調べると証明書のバージョンが低いことが原因であった。SSL のライブラリをバージョンアップし再び証明書を作成したところ, 認証でき暗号化もできていた。

(※文責: 水澤吉博)

プログラム

サービスを実現するにあたって必要となる機能は, メール受信時に Selfi が貸出可能かを判定すること, 貸し出し時間の超過を確認すること, サービスの停止, サービスの再開などがある。また, 何らかの障害が発生した場合に備えて, データベースのバックアップを取る必要もある。今回は, 表 5.4 のような Perl のプログラムを作成した。以下に各プログラムの詳細を記す。

表 5.4 プログラムの機能

プログラム名	機能
receive.pl	メール受信時に動作し, Selfi の利用状況を確認し, 貸出可能ならログイン用のパスワードを記載したメールを返信し, 貸出が不可能ならその旨を記載したメールを送信する。
check_receive_time.pl	貸出予約が完了してから 30 分後に起動し, 貸出予約の状態が続いている場合, 予約のキャンセルを行う。
check_rental_time.pl	貸出時刻から 1 時間 30 分以上貸出が継続している場合, ユーザに警告のメールを送信する。
stop_service.pl	サービスの停止を行う。これを実行すると利用申請のメールを送っても, 貸出が不可能な状態となる。
start_service.pl	サービスの再開を行う。stop_service.pl を実行した後に再開したい場合実行する。
sqldump.pl	1 日に一度 crontab ^{*1} によって起動される。データベースのバックアップを行う, 3 つ以上バックアップがある場合古いものを削除する。

*1 コマンドの定時実行を行う。

receive.pl

概要 receive.pl の最も基本的な動作は、利用可能な Selfi が存在する状態で、ユーザが利用申請のメールを送信すると、貸出予約が完了するというものである。

貸出予約とは、実際に Selfi に乗車するのに必要となるパスワードが発行された状態で、このパスワードを Android 端末に入力することで、貸出が完了する。貸出予約の状態になった時、つまり、パスワードが発行された時から 30 分経過しても、貸出が完了しない場合には、貸出予約の状態を破棄し、発行したパスワードを利用不可能にする。また、貸出予約の状態のユーザが再度空メールを送信した場合にも、貸出予約の状態を破棄する。

Selfi が貸出不可能である場合には、その旨のメールをユーザに送信する。

詳細 /etc/aliases 内で、

```
user: "|/usr/bin/perl perl_program.pl"
```

のように設定することによって、Postfix に user 宛のメールが届いた時に、perl_program.pl を起動し、標準入力にメールが渡される。これによって、メールを送信してきたユーザのメールアドレスを取得することが可能となる。

Selfi が貸出可能かの判断は、以下の条件から決定する。

1. 現在時刻が貸出時間内であるか。
2. ユーザがブラックリストに登録されているか。
3. ユーザが現在貸出予約中であるか。
4. ユーザが現在貸出中であるか。
5. 貸出可能な Selfi が存在するか。

これらの内、どれかに問題がある場合には、プログラムは条件ごとに貸出不可能な旨を記載したメールを送信して終了する。

receive.pl は、上記の 5 つの条件全てに該当しなかった場合に、貸出予約の処理を続行し、貸出予約を完了するのに必要となる情報である、貸出番号、ユーザ ID、Selfi ID、パスワード、メール受信時刻をデータベースに挿入する。そして、Selfi の状態をパスワードの入力待ちの状態にする。これらの処理が正しく行われた後、ユーザに貸出予約が完了した旨のメールを送信する。

最後に、貸出予約から 30 分経過しても、貸出が完了しない場合を検出するために、30 分後に確認を行うプログラムの起動を予約する。具体的には、system() を用いて、at コマンド *2 を現在時刻の 30 分後に設定し、プログラムを登録する。

check_receive_time.pl

貸出予約が完了してから、30 分後に at コマンドによって実行され、貸出予約が継続しているかを確認するプログラム。貸出予約の状態が続いていた場合には、予約をキャンセルし、ユーザにキャンセルした旨のメールを送信する。貸出状態になっていた場合には、データベースに確認をするだけで何も行わない。

*2 指定時刻に一度だけ job を実行する。

check_rental_time.pl

概要 check_rental_time.pl は、ユーザの貸出時間の超過の有無の監視を行う。基本的に、このプログラムは常に、at コマンドの job に登録されており、いつ起動するかは、貸出状況によって変化する。

詳細 貸出がない場合、check_rental_time.pl は貸出時間の最大値である 1 時間 30 分間隔で起動する。

貸出がある場合、その貸出の現在時刻が、貸出時刻から 1 時間 30 分超過している場合には、その Selfi を使用しているユーザに対して返却を促すメールを送信し、管理者のアドレスに貸出時間の超過を検知したという旨のメールを送信する。check_rental_time.pl が起動するたびに、一度返却を促すメールを送ったユーザ宛にメールを送信してしまうと同様の内容のメールが何度も送信されることとなるので、一度メールを送信したユーザに対しては、返却が完了するまで、二度とメールを送信しないようにしている。

貸出がある場合の次回起動時刻は、現在の貸出の中で、貸出時刻の最も早いものに 1 時間 30 分加算した時刻となる。

貸出の終了時刻に起動するようになっており、次回の起動時刻を、曜日、祝日によって決定する。具体的には、休日でない日の貸出開始時刻の 1 時間 30 後に起動するようになっている。

stop_service.pl

サービスの停止を行う。貸出サービスに何らかの障害が発生し、Selfi の使用が不可能である場合に実行する。貸出予約中のユーザがいた場合には、貸出が不可能になった旨のメールを送信する。コマンドライン引数に、Selfi の ID を入力することで、任意の Selfi の停止が可能となっている。

start_service.pl

サービスの再開を行う。基本的に stop_service.pl を使用した後にしか使用しない。stop_service.pl と同様に、コマンドライン引数に Selfi の ID を入力することで、任意の Selfi の再開が可能となっている。

sqldump.pl

このプログラムは crontab に登録されており、1 日に一度、指定した時刻に起動される。データベースのバックアップを行う。バックアップは 3 日分保管されることとなっており、障害が発生した場合に、3 日前まで遡ってデータを復旧することができる。

(※文責: 吉田豊)

5.1.3 マイコン

以下に、4.1.3 節で述べた課題解決のために、実際にマイコンを用いて行った課題解決のプロセスを示す。

- Arduino への電力供給方法の確立.
- Selfi のロック機能の実装.
- バッテリー残量計測機能の実装.
- 速度計測機能の実装.
- Android 端末との無線通信機能の実装.

これらの各項目について、以下に詳細を述べる。

Arduino への電力供給方法の確立

Arduino の入力電圧 本プロジェクトで私たちは Arduino の中でも Arduino Mega というマイコンを使用した。この Arduino への電源供給を行うには、Selfi のバッテリーから供給する方法がある。しかし、このバッテリーは最大 26V と電圧であり、Arduino には電圧が高すぎると考えた。Arduino の電源供給をどのように行うか検討した結果、去年のプロジェクトと同様に Selfi のモータードライバーから供給しようと考えた。このモータードライバーによっては Selfi のバッテリーの電圧を 5V まで下げ、使用することができた。

しかしその後、電圧の値を Arduino で正確な値を測ることができないという問題が発生した。この問題の原因は、Arduino Mega の入力電圧が 5V では足りないということだった。今回使用した Arduino Mega は動作電圧は 5V であるが、Arduino Mega の内部抵抗によって基準電圧が下がってしまうためであった。そのため、Arduino への電源供給には 7V から 12V の電圧が必要であった。

レギュレータでの Arduino の電源供給 もう一度検討を行った結果、図 5.10 の 9V1A 3 端子レギュレータを用いることで、Selfi のバッテリーの電圧を 9V まで下げ、Arduino に電源供給することとした。また、バッテリーの電圧が高いのでショートしてしまう可能性を考え、バッテリーからレギュレータまでの間に図 5.11 のヒューズを挟んでつなげることにした。外部電源で Arduino に電源供給するため、DC ジャックを使うことにした。その結果、Arduino によって正確な値を測ることが可能となった。

レギュレーターの排熱問題 Arduino の使用中、正常に作動しなくなるという問題が発生した。Arduino が起動して間が空いてから問題が発生することとレギュレータから発せられる熱から、原因はレギュレータがにあると考えた。レギュレータは入力と出力の電圧差を熱に変えて放出している。本プロジェクトで使用しているレギュレータは最大 17V の電圧差が生じるため、発熱量が大きくなりすぎ熱処理が追いつかなくなり、うまく作動できなくなったのではと考えた。その排熱効率を上げるためレギュレータに図 5.12 のようなヒートシンクを取り付けた、ヒートシンクとは、レギュレータに取り付けることで空気に触れる表面積を増やし、排熱効率を上昇させる装置である。ヒートシンクをつけ、レギュレータを稼働させ続ける実験を行った。その結果、最低でも 10 時間の間、この排熱問題が生じることはなくなった。本プロジェクトの目標である公立はこだて未来大学内でのレンタルサービスの利用予定時間から考えて十分な時間使用できると判断した。

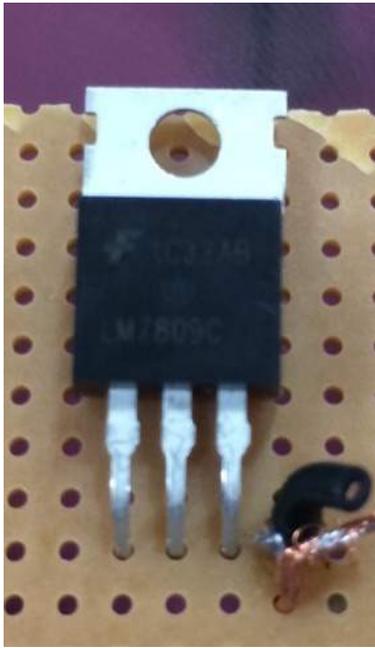


図 5.10 9V1A3 端子レギュレータ

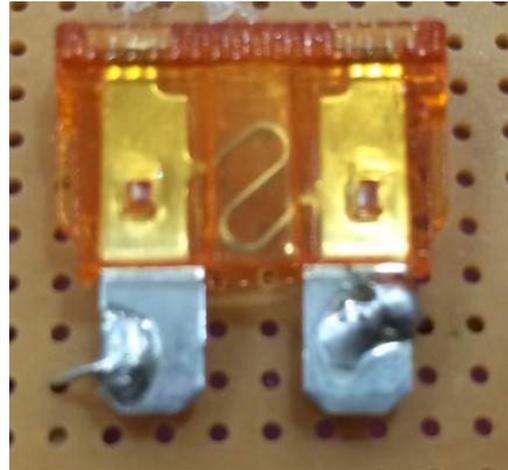


図 5.11 ヒューズ

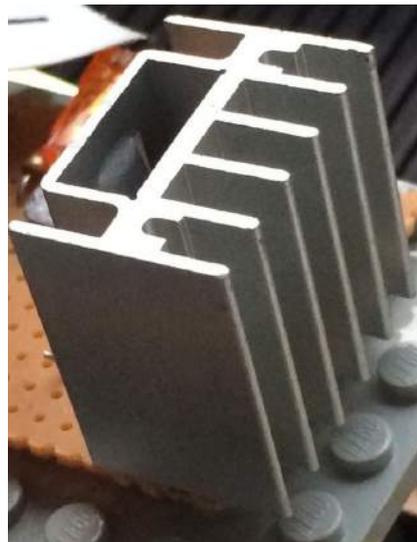


図 5.12 ヒートシンク

(※文責: 山本一希)

Selfi のロック機能

使用した電子部品 Selfi のロック機能を実現するために、リレースイッチを用いた。リレースイッチとは、電磁石を用いることで電気信号によってスイッチを ON/OFF させることができる電子部品である。本プロジェクトでは図 5.13 に示した Arduino に取り付けて使うリレーシールドを用いた。

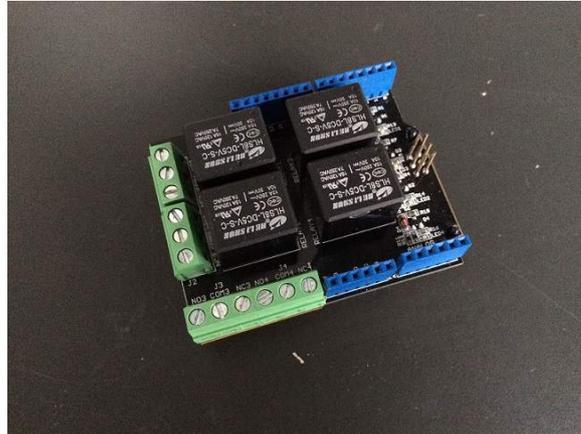


図 5.13 リレーシールド

リレースイッチの取り付け箇所 リレースイッチを用いて Selfi をロックするために、まずは Selfi の制御に関する回路の中で、どの部分にスイッチを入れるべきかについて検討を行った。その結果、Selfi の制御を入れる際に用いる、ハンドル部分のスイッチの代わりとしてリレースイッチを用いることとした。ハンドルのスイッチを押しながら Selfi を水平の状態にすることによって Selfi に制御が入るため、これをリレースイッチに変えることによって、Arduino から Selfi に制御を入れられる状態と入れられない状態を切り替えることができる。以上の Selfi の状態の切り替えによって Selfi のロック機能を実現することとした。また、「スイッチを押しながら Selfi を水平にする」という動作を簡略化でき、Selfi の乗りやすさの向上も期待できると考えた。

リレースイッチを切り替えるタイミング Selfi は Android アプリにログアウトしている状態ならばロックする、すなわちリレースイッチを OFF にし、ログインしている状態ならばロックを解除する、すなわちリレースイッチを ON にする必要がある。しかし、Selfi の制御が入っている状態でリレースイッチを ON にしたままにすると、Selfi に乗っている際に危険な挙動をする可能性も考えられたため、ログインしていても Selfi の制御が入っている状態の時にはロックする、すなわちリレースイッチは OFF にすることとした。以上のことを表 5.5 にまとめて示す。

表 5.5 Android アプリと Selfi の状態の対応関係

Android アプリの状態	Selfi の制御	Selfi のロック状態	リレースイッチの ON/OFF
ログイン	ON	ロック	OFF
	OFF	アンロック	ON
ログアウト	ON	ロック	OFF
	OFF		

リレースイッチの切り替えに必要な情報の取得 上記のタイミングでリレースイッチを切り替えるために、以下の情報を取得する必要がある。

- Android アプリのログイン情報
- Selfi の制御情報

まず、Android アプリのログイン情報は、Bluetooth を用いた無線通信によってアプリから受信する。具体的な通信の方法については後述する。アプリからは 0 か 1 でログイン情報が渡され、0 であればログアウト状態、1 であればログイン状態であると判断する。

次に、Selfi の制御情報は Selfi のハンドルに取り付けられた LED の電気信号から判断することとした。Selfi のハンドルの LED は Selfi に制御が入っている際に光るようになっているため、これを制御の ON/OFF の状態を判別するために用いることができると考えた。また、この LED は PWM 信号によって明るさを制御していたため、Arduino にアナログ信号として入力し、その値が 200 以上であれば LED は ON、200 未満であれば LED は OFF であるというように閾値を定め、LED の状態を判別することとした。

これらの情報を元に、アプリから 1 が渡され、かつ Selfi のハンドルの LED が ON であればリレースイッチを ON とするプログラムを作成した。

(※文責: 西村祐輝)

バッテリー残量計測機能

バッテリーの分圧 バッテリー残量計測機能を実現するために、分圧という方法を利用することとした。分圧とは図 5.14 のよう抵抗を使って並列回路を作ることによって、電圧を分割させるという技術である。



図 5.14 分圧回路

電源装置による分圧実験 分圧によってどの程度まで正確に電圧を計測できるかを知るために実験を行った。電源の電圧を $E[V]$ 、Arduino の電圧を $V[V]$ 、Arduino で計測した値を X とする。電源部分はバッテリーの代わりに図 5.15 のような最大電圧 12V 電源装置を 2 つ直列につなげることで最大電圧を 24V とし、図 5.14 のような回路で行うこととした。Arduino で計測できるのは最大 5V までであり、Selfi のバッテリーの最大電圧は 26V なため、計算が容易なようにバッテリーの抵抗値を図 5.14 のように抵抗を 5k Ω と 21k Ω とした。並列で抵抗の間につながっている Arduino の電圧 V は以下ようになる。



図 5.15 電源装置

$$V = \frac{5}{26}E \quad (5.1)$$

また，Arduino はアナログ値で計測しているため， V の値が $5V$ であった時， X の値は 1023 となる．よって， X の計算式は以下ようになる．

$$X = \frac{1023}{5}V \quad (5.2)$$

よって，式 (5.1) を式 (5.2) に代入して， X は以下のように表せる．

$$X = \frac{1023}{26}E \quad (5.3)$$

E の値が 2 つの電源装置の最大電圧 $24V$ だとすると，式 (5.3) より X は約 944 となる．しかし，実験を何回か繰り返したが，結果は X の値が予想の値を平均 100 ほど大きくなってしまった．この値は式 (5.3) より E の値の $2.54V$ の誤差が出ていることがわかった．この誤差は E の値の約 10% なので，誤差が大きすぎると考えた．そのため，その誤差の原因を探し，小さくする方法を考える必要があった．

誤差の原因の検出する実験 まず，誤差の原因が Arduino で読み取った値がデータ上ずれてしまっているのではと考えた．そのため，読み取った値を右シフトを行った．例えば， X の値が 944 であり， X の値は 2 進数で表すと以下ようになる．

$$X_2 = 1110110000 \quad (5.4)$$

この値を 2 つ右シフトを行った値を X' とした場合，以下のように表せる．

$$X'_2 = 11101100 \quad (5.5)$$

この時， X' の最大値は 255 となる．その結果， X' を 10 進数で表すと，以下ようになる．

$$X' = 236 \quad (5.6)$$

実験では 2 つから 4 つまで右シフトを行った。しかし、右シフトを行っても、誤差の値は変わらなかった。

次に、前項の「Arduino への電力供給方法の確立」で述べたように、Arduino の電源供給の電圧が低すぎたことが原因ではないかと考えた。Arduino への電源供給を 5V としてしまうと、内部抵抗によって内部電圧が下がり、その結果 Arduino の基準電圧が下がってしまうからである。基準電圧が下がると、その下がった値から 5V の間でも、 X の値が 1023 となってしまう。例えば、基準電圧が 4.5 となってしまうとする。この時、 V の値が 4.5V から 5V の時に X の値は 1023 となる。 E の値が 24V である時、式 (5.1) より V の値は約 4.6V となり、 X の値が 1023 になってしまう。その結果、79 の誤差が生じてしまう。その後、前項の「Arduino への電力供給方法の確立」で述べた方法で問題を解決し、実験を行ったところ、 X の値は約 10 の誤差が生じた。この値は式 (5.3) より E の値の 0.25V の誤差が出ていることがわかった。この誤差は E の値の約 1% なので、ほぼ誤差なく電圧を計測できることがわかった。実際に使用する Selfi のバッテリーを用いた実験でも同様の結果が得られた。

Selfi の稼働電圧 Selfi のバッテリー残量の計測を行うためには、Selfi の稼働電圧の最低値と最大値を測る必要があった。そのため、Selfi の動きが悪くなり充電が必要となった際、バッテリーの電圧を計測すると約 24V であった。最大値はバッテリーが最大 26V なので Selfi のバッテリー残量は $E = 26[V]$ の時は 100%、 $E = 24[V]$ の時は 0% となる。よって Selfi のバッテリー残量を Y とすると、計算式は以下のようになる。

$$Y = (E - 24) \frac{100}{26 - 24} \quad (5.7)$$

$$Y = 50(E - 24) \quad (5.8)$$

また、式 (5.3) より、以下のように表せる。

$$Y = 50 \left(\frac{26}{1023} X - 24 \right) \quad (5.9)$$

この値より、Selfi のバッテリー残量を計算することができた。

(※文責: 山本一希)

速度計測機能

使用したセンサ Selfi の速度を計測するために、図 5.16 に示したフォトリフレクタと呼ばれるセンサを用いた。フォトリフレクタとは、発光素子（赤外線 LED）と受光素子（フォトランジスタ）を 1 つにしたセンサであり、発光素子から放出した光が物体に当たることで反射し、その反射光を受光素子によって検出することによって物体の有無や距離を測ることができるセンサである。このフォトリフレクタを用いて Selfi の速度を計測する。

計測対象 速度を計測するために、フォトリフレクタを用いて Selfi のタイヤの回転数の計測を行う。タイヤの回転数を測ることで、一定時間に Selfi の進んだ距離を求めることができ、そこから平均の速度を求めることができる。また、Selfi は左右のタイヤの回転数に差をつけることで曲がるため、片方のタイヤの回転数のみの計測では正確な速度が得られないと考えた。そこで、左右のタイヤの回転数を別々に求め、平均を取ることで正確な速度を求めることができると考えた。

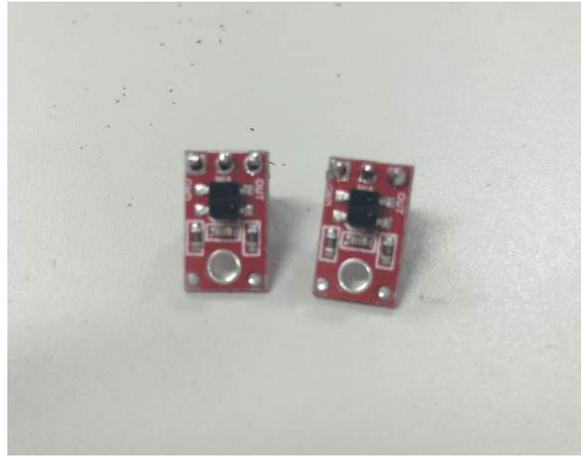


図 5.16 フォトリフレクタ

タイヤの回転数の計測方法 Selfi のタイヤの回転数を計測するために、Selfi 本体にフォトリフレクタを取り付けた。初めはタイヤのスポークをフォトリフレクタで検出することによって、その検出回数をカウントし、回転数を計算するという手法を用いた。しかし、スポークが細く、フォトリフレクタで検出できないことが多々あり、正確な速度を求めることができなかった。そこで、図 5.17 に示した白と黒のマーカを用いし、スポークとスポークの間に設置することとした。フォトリフレクタは光の反射によって物体の位置を検出する。よって、反射率の高い白は近くにあるように検知し、反射率の低い黒は遠くにあるように検知する。この差によってマーカの数を検出し、回転数を求めることとした。



図 5.17 タイヤに取り付けたマーカ

速度の計算方法 上記の方法で求めたマーカーの検出回数から Selfi の速度 $V[\text{km/h}]$ を計算する手順を以下に示す．ここで，計測時間は $T[\text{s}]$ とする．

1. マーカーの検出回数から回転数を求める．左のマーカーの数を M_l ，左のマーカーの検出回数を S_l ，右のマーカーの数を M_r ，右のマーカーの検出回数を S_r とすると，左のタイヤの回転数 N_l と右のタイヤの回転数 N_r はそれぞれ次のように求められる．

$$N_l = \frac{S_l}{M_l} \quad (5.10)$$

$$N_r = \frac{S_r}{M_r} \quad (5.11)$$

2. よって，式 (5.10) と式 (5.11) より，左右のタイヤの平均の回転数 N は以下のように求められる．

$$N = \frac{N_l + N_r}{2} \quad (5.12)$$

3. ここで，タイヤの半径を $R[\text{m}]$ とすると，タイヤの円周 $C[\text{m}]$ は以下のように求められる．

$$C = 2\pi R[\text{m}] \quad (5.13)$$

4. よって，式 (5.12) と式 (5.13) より，Selfi の進んだ距離 $D[\text{m}]$ は以下のように求められる．

$$D = NC[\text{m}] \quad (5.14)$$

5. したがって，式 (5.14) より，計測時間 $T[\text{s}]$ の間の Selfi の平均の速度 $v[\text{m/s}]$ は以下のように求められる．

$$v = \frac{D}{T}[\text{m/s}] \quad (5.15)$$

6. 最後に，以下のように $v[\text{m/s}]$ から $V[\text{km/h}]$ に単位を直す．

$$V = v * 3.6[\text{km/h}] \quad (5.16)$$

以上のようにして求めた速度を Android アプリに送信し，マーカーの検出回数をリセットして再び $T[\text{s}]$ の間回転数の計測を行い，再び計算をすることで速度を一定間隔で更新し続けることを可能とした．

(※文責: 西村祐輝)

Android 端末との無線通信機能

使用した電子部品 Arduino と Android 端末との無線通信に Bluetooth を用いた。中でも特に、BLE と呼ばれる規格の Bluetooth を用いて通信を行った。本プロジェクトでは RedBearLab より販売されている、図 5.18 に示した BLE シールドを用いた。



図 5.18 RedBearLab BLE シールド

RedBearLab BLE シールドの機能 このシールドでは、同社より提供されている「nRF8001」という Arduino のライブラリを用いて開発を進めることが可能である。このライブラリで提供される関数の内、以下に示したものをを用いて開発を行った。

ble_set_name(String)

BLE のデバイス名を変更する関数。BLE のデバイス名を特定のものに変更することで、Selfi に設置した Android アプリ以外からの誤ったアクセスを防ぐ。

ble_begin()

BLE の通信を開始する関数。

ble_available()

BLE に送られたデータのバイト数を返す関数。受信したデータを受け取る際に、この関数の戻り値が 0 でなければ、データを受信したと判断する。

ble_read()

BLE に送られたデータの先頭の 1 バイトを返す関数。Android アプリから送信されたデータを読み込む。

ble_write(byte)

引数で指定された 1 バイトのデータを送信バッファに格納する関数。Android アプリに送信するデータを送信バッファに格納する。

ble_do_events()

送信バッファに格納されたデータの送信を行う関数。実際に Android アプリに対してデータを送信する。

ログイン情報のデータ受信 Android アプリから送信されるログイン情報のデータを取得する。Android アプリにログインした際には 0, Android アプリからログアウトした際には 1 の 1 バイトのデータが送信される。ble_available 関数によってデータの受信を判断し, ble_read 関数によって実際にデータを読み込む。この受信したデータを用いて, Selfi のロック機能を実現する。

バッテリー残量のデータ送信 計測したバッテリー残量のデータを Android アプリに送信する。この際, Arduino から送信するデータ量を減らすために, 計測したバッテリー残量を 0% から 100% の値に変換して送信することとした。Selfi の仕様書より, バッテリーの最大電圧は 26V, Selfi が走行可能な最小電圧は 24V であり, アナログ値では 944 から 1023 の範囲である。よって, 計測したバッテリー残量を map 関数を用いて 944 から 1023 の範囲から 0 から 100 の範囲に変換することによって百分率への変換を行う。int 型のバイト数は 2 バイトであるが, 最大値が 100 であれば下位 1 バイトのみを送信すればよい。したがって, 変換したデータを 0xff との論理積をとり, 下位 1 バイトのみを ble_write 関数によって送信バッファに格納し, ble_do_events 関数によって送信する。

速度のデータ送信 計測した速度のデータを Android アプリに送信する。この際, 計算した速度の値は float 型であるが, float 型は 4 バイトであるため, そのまま送信するとデータ量が大きくなってしまふ。そこで, 計算した速度の値を 100 倍し, int 型にキャスト変換することで小数第 2 位までの精度で送信することとした。Android アプリでデータを受信した後に 100 で割ることで元の速度の値に戻して利用するという仕組みとした。また, 送信する速度の値は 100 倍すると 2 バイト分の int 型のデータとなるため, 上位 1 バイトと下位 1 バイトで分けて送信する必要がある。そこで, まずは int 型にした速度のデータを右に 8 ビットシフトし, 0xff との論理積をとったものを ble_write 関数によって送信バッファに格納する。次に, int 型にした速度のデータをそのまま 0xff との論理積をとり, ble_write 関数によって送信バッファに格納する。これにより, 上位 1 バイトと下位 1 バイトを別々に送信バッファに格納する。これを ble_do_events 関数によって送信することで速度のデータを Android アプリに送信する。

(※文責: 西村祐輝)

5.2 ハードウェアの課題解決プロセス

以下に、4.2 節で述べた課題解決のために、実際に行った Selfi 本体に対する課題解決のプロセスを示す。

- Selfi の組み立て.
- 使用機材の検討.
- 安全装置の作成.
- Android 端末を Selfi に取り付けるためのケースの作成.
- Arduino を搭載するための回路製作.

これらの各項目について、以下に詳細を述べる。

Selfi の組み立て

この項目では Selfi 本体の組み立てと調整作業について説明する。Selfi は本来完成品と組み立てキットの 2 種類で販売されているが、Selfi 自身の仕組みを理解するために Selfi の組み立てキットを購入しそこから Selfi を作成した。今年作成した Selfi 以外にも前年のプロジェクトにおいて作成した Selfi が 2 台存在している。Selfi は大きく分けてドライブユニット、メインフレーム、バッテリーホルダ、ハンドルユニット、基板の 5 個の要素からできている。これらのパーツを完成させて最終的に配線をつなぎ、パーツを合体させることで Selfi が完成する。図 5.19 の左右にあるタイヤ部分がドライブユニット、中心にあるのがメインフレーム、下にあるのがバッテリーホルダ、上にあるのがハンドルユニットとなっている。

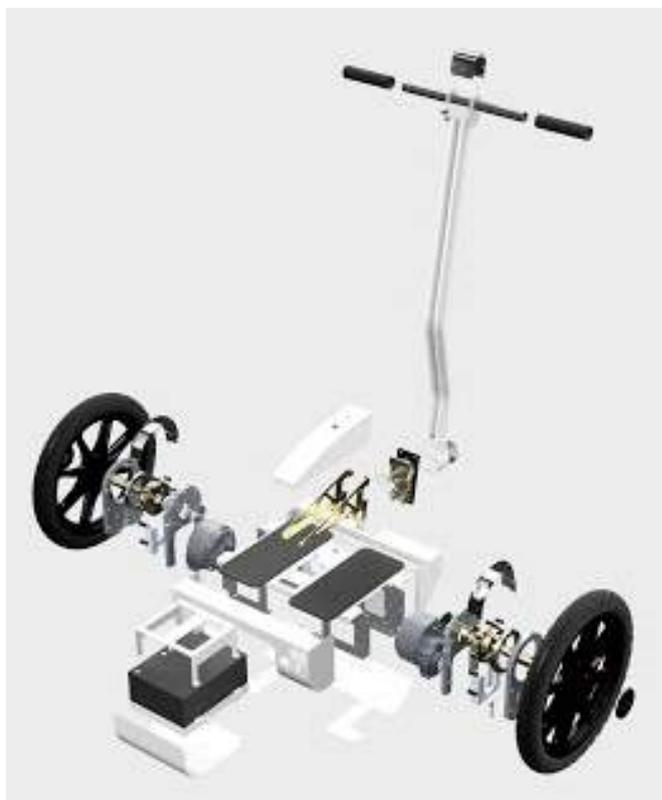


図 5.19 Selfi の構造 [1]

Make Use of Personal Mobility for Indoor Movement

基板部分は目に見えるところにはないが、Selfi の制御や速度、ハンドルの傾き等のセンサー類を管理している mbed が本体内部にある。図 5.20 が実際の mbed である。この mbed には最初からプログラムが組み込まれていて、プログラム自体はそのまま使っている。またこの mbed には予めスピーカが取り付けられており、一定の速度を超えた時点で警告音がなったり、mbed と PC を繋いだ時に音になるようになっている。この mbed で各種センサー類の値を読み取り、モータードライバの方へパルス波を送り速度を調整している。この基板を作成するために私達ハード班は半田付けの技術も習得した。

また、ドライブユニットをひと通り組み立てた後、電源装置を用いて実際に電流を与えてモータを回してみたところチェーンから異音が発生した。チェーンの弛さを微調整し、何度か組み立てと解体を繰り返した。最終的に回転速度を上げてチェーンを慣らしてみたところ異音がかなり小さくなったため、そのまま他の部分の組み立てに移った。

他のハンドルユニット、メインフレーム、バッテリーホルダはパーツを組み立てるだけの作業だったので組み立てていく上で問題は発生しなかったが、各パーツの最終組立作業に移ると色々なところに設置したセンサやバッテリーのケーブルなどをつなげるためにケーブルを通すスペースの問題が生じた。バッテリーを詰め込んだ Selfi 内部は非常に狭くバッテリーでケーブルを踏まないように、また今後他のものを搭載する可能性も考慮し、ケーブルが邪魔にならないように慎重に作業を行った。

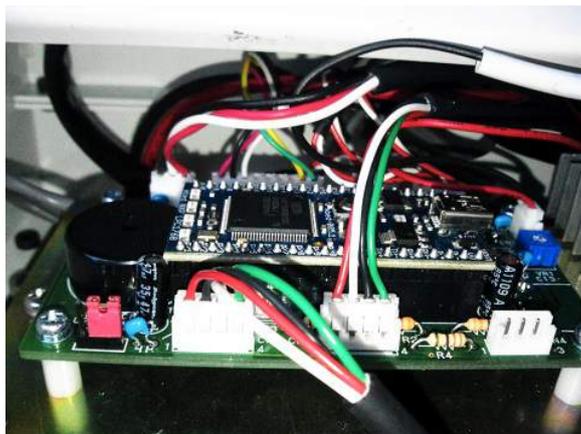


図 5.20 mbed

次に表面上の組み立てがある程度完成した後に Selfi のキャリブレーションの調整を行った。これを行うことにより、Selfi の左右のタイヤの回転数を調整し、ハンドルの傾きできちんと曲がることのできるようになる。先に述べた通り、Selfi の制御は mbed で行っている。Selfi には大きく分けて足が Selfi に付いているかどうか判断する圧力センサ、Selfi 本体の傾きを測定するジャイロセンサ、ハンドルの傾きを測定するステアリングのセンサの 3 つのセンサが取り付けられている。勘違いしやすいところであるが、Selfi は体重を掛けた圧力で速度を管理しているのではなく、Selfi 本体の傾きを測定することにより速度を管理している。圧力センサは制御を入れるときに人が本当に乗ろうとしているかどうかを判断するためにしか働いておらず、圧力がかかっているか、もしくはかかっていないかの二種類しか判断していない。図 5.21 はキャリブレーションの画面を示している。この状態では値が検出されていないが各パラメータにそれぞれの値が表示される。offset の値を調整することにより Selfi の水平状態を 0 とすることで初期値を設定する。また、モータの PWM 幅も調整することにより左右のモータの回転数を設定することが可能となっている。

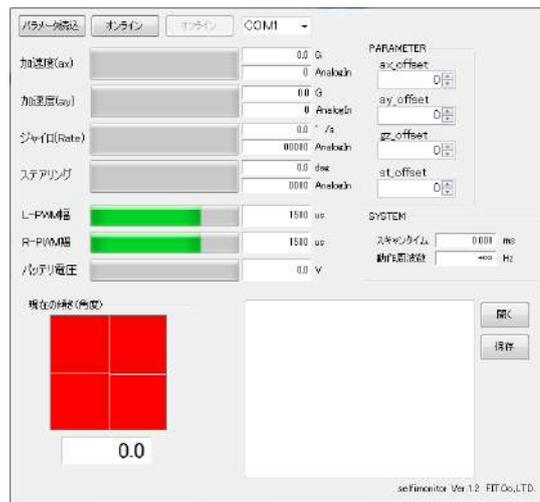


図 5.21 キャリブレーションの画面

(※文責: 稲葉祐太)

Selfi の調整

Selfi は完成させることができたが、Selfi の元々の欠点として、タイヤのチューブの耐久性が低いことがあった。なので、別途より頑丈なタイヤのチューブを購入しそれを用いた。また、空気の入れ方もきちんと行わなければチューブが傷ついてしまう可能性もあるので、正しい方法を学んだ。それ以外にもハンドルの取っ手の中心部についてある制御スイッチを強く押してしまうことによりボタンが陥没してしまう問題もあった。この問題はボタンの裏の部分をテープで固定することによりある程度防ぐことが可能であったが、完全に防ぐことが不可能であったため、Selfi の制御を完全に Android アプリ側で管理することにより解決することができた。また、制作を進めていく上で Selfi の制御が唐突に解除されてしまう現象が発生した。色々と実験をしていくうちにその現象はバッテリー残量がある程度減少した状態、且つ急加速や急減速を行いモータに過度の負担を掛けた場合に発生することが判明した。その際にすでに搭載されていた Arduino への電力供給により発生している可能性があったが、Arduino を外した状態でもその現象が発生したため、Arduino はこの問題とは無関係であることも判明した。その現象を再現して各部分を観察してみたところ、制御が落ちる瞬間に mbed 側への電力供給が止まってしまっていることが判明した。元々バッテリーから直接電力が mbed 側に流れているわけではなく、一度、図 5.22 に示したモータドライバに電力を送りそこから電圧を下げ mbed 側に送っているためモータドライバ側も確認してみたところモータドライバへの電力供給も止まってしまっていることがわかった。バッテリーとモータドライバ間の電線には問題がなかったためこの現象が起こった原因はモータドライバに過度な電流を掛けてしまうことで電流が止まってしまうことにあると判明した。同型のモータドライバは国内で販売されておらず、すぐに取り寄せるのには時間がかかるので他の正常に動く Selfi のモータドラ

Make Use of Personal Mobility for Indoor Movement

イバと交換してみたところ正常に動いたのでそのまま運用していくことにした。交換元の Selfi のモータドライバと今回破損した Selfi のモータドライバの他にもう一台に搭載されているモータドライバもあるが、そちらの方にも少々問題があり、モータの電流のプラス、マイナスの切替時に動作が非常に不安定になる状態であった。こちらのモータドライバの方も交換することにより解決したのでこの問題もモータドライバが原因であると判断した。

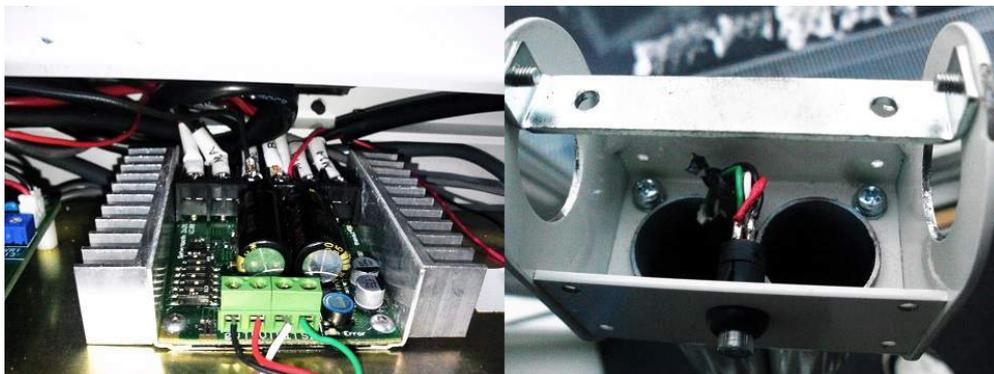


図 5.22 左：モータドライバ，右：ハンドルボタン

(※文責: 稲葉祐太)

使用機材

レーザーカッター

安全装置や Android ケースを作成するにあたって、パーツの切り出しを行うためにレーザーカッターを使用する。レーザーカッターを使用するためには、レーザーカッターの講習を受け、簡単な小テストに合格する必要がある。この工程を行い、レーザーカッター講習終了証を発行することでレーザーカッターの使用が許可される。レーザーカッターでは、主にアクリル板を素材として切り出した。安全装置については厚いアクリル板を、Android ケースについては薄い物を材料として使用した。レーザーカッターを使い慣れないうちは、パーツを切り出すのにアクリル板の広い部分を使って少ないパーツしか切り出すことができなかったが、繰り返し使用していくことで狭いスペースでも思い通りのパーツを切り出すことができるようになり、材料を有効に使うことができるようになった。

(※文責: 中田智子)

CAD

安全装置のパーツや Android ケースの、作成にあたって DraftSight というソフトを使って設計図を作成した。DraftSight では、主にキーボードを使ってパーツの形を作っていくため、精密な設

計を行う作業に適していた。また、レーザーカッターを使用するときに使うファイル形式にも変換できることから、今回のパーツの切り出しに使用することとした。CAD データは、前年度のものを参考にしつつ今年度使用するものに合わせて設計し直した。DraftSight を使って設計する上で、はじめのうちは操作に慣れず簡単な形しか作ることができなかった。しかし、何度も設計を繰り返して調整を加えていくことで自分の思い通りの設計図を作成できるようになった。

(※文責: 中田智子)

安全装置

安全装置とは、搭乗の際に後に動かないようにある程度抑制を加える装置のことである。搭乗の際に後ろにある程度の抑制を加えるだけで初心者が Selfi に乗るときに安全に乗れるようにサポートする。できるだけ簡単な操作で安全装置をセットし、乗り降りに邪魔にならないものを想定している。安全装置では、初心者によくありがちな搭乗の際に起こるぐらつきを軽減することを目的とする。

素材と加工方法 安全装置の作成にあたり、レーザーカッターとアクリル板を使用することとした。素材としてポリカーボネートという強度のあるものを使う予定であったが、レーザーカッターではポリカーボネートを加工することができないことが判明したため、強度の問題はあるもののレーザーカッターで加工することのできるアクリル板を使用することとなった。設計に関しては DraftSight を使用し、CAD データを作成することとした。安全装置の取り付け予定場所の寸法を計測し、それに合った設計図を作成した。安全装置は、人が乗っている状態で使用する。このとき Selfi 本体に傾きが発生するため正確な寸法が取れず何度か設計をやり直しや、調整が必要となった。作成する上で安全装置の取り付け場所に金具で取り付けることができなかったため、強力粘着テープで取り付けてテストした。実際に作成した安全装置は、テスト用に作成した試作品と機構の確認が取れた完成品がある。

試作品の製作 まず、テスト用に作成された安全装置について記述する。この安全装置を安全装置 1 号と呼ぶことにする。安全装置 1 号の図を図 5.23 に示す。図 5.23 のような装置をレーザーカッターと CAD を使って作成した。図の手前が Selfi に取り付けられた時の前の部分で、奥が後ろの部分となる。構造として、Selfi 本体と接点を持つ箇所は 2 か所である。この安全装置の設置場所は、Selfi 本体の下部分である。鋭利な凹凸部分が地面に向かってような形で取り付ける。この鋭利な凹凸のついたパーツのことを摩擦パーツと呼ぶことにする。Selfi 本体と接点を持つパーツにはばねを付け、安全装置セット後とセット前ではばねの伸縮が変わるように設計した。ここで Selfi 本体と接点を持つパーツを変動パーツと呼ぶことにする。安全装置セットした状態ではばねが伸びた状態になり、前に進むとばねが少しずつ縮むためその力を利用して、安全装置が外れるような構造である。また、後ろに動くことないように変動パーツには、摩擦パーツと接合しているねじが動くことのできる範囲を制限するような構造にした。また、変動パーツに Selfi 本体にぶつかって動かないようにする構造となっている。Selfi 本体と、変動パーツは動きに支障が出ないように設計し、安全装置を使用しない状態でも走行に邪魔になることのないように、できるだけコンパクトになるように設計した。これらの変動パーツと摩擦パーツの組み合わせたものを二組用意し、つなげたものが安全装置となる。



図 5.23 安全装置 1 号

試作品から判明した問題点 しかし、安全装置 1 号ではあまりうまく動かなかった。これには二つの原因が考えられた。まず、1 つ目は部品に対してちょうどよい長さ、強さのばねがなく十分なばねの伸縮の力が発揮されなかったことがあげられる。ばねを取り付ける実際の長さは 4.5cm 程度だったが一般に販売されているばねでは対応することができなかった。これによって、ばねの力で跳ね上がるという部分がうまくいかなかった。2 つ目の原因として、摩擦パーツと変動パーツの接合部分のバランスが悪かったことがあげられる。バランスが悪くどちらかに傾いてしまうため地面に接触してしまうことがあった。結果、引きずってしまうので走行の邪魔となった。これら 2 つの原因から、安全装置 1 号はうまく動作しなかった。しかし、Selfi 本体下の狭い空間でも安全装置の取り付けが可能であり、後ろへいかなないように抑制をこの構造で行えるということ確認できた。

完成品の製作 次に、機構の確認が取れた完成品の安全装置について記述する。この安全装置を安全装置 2 号と呼ぶことにする。安全装置 2 号は、安全装置 1 号のうまく動作しなかった原因を参考に作成し直すこととなった。安全装置 2 号を図 5.24 に示す。図 5.24 のような装置をレーザーカッターと CAD を使って切り出し、組み立てた。構造としては、安全装置 1 号とほとんど同じであるが、Selfi 本体と接点を持つ箇所は 4 か所である。この安全装置の設置場所は、Selfi 本体の下部分で、摩擦パーツが地面に向かってような形で取り付ける。変動パーツは 4 つに増え、それぞれにはばねを付け安全装置セット後とセット前ではばねの伸縮が変わるように設計した。安全装置 1 号で使用したばねよりも少し短い長さのばねを使用した。安全装置 1 号では、後ろに進まないにするための抑制の役割をするねじが動くことのできる範囲をばねと地面が垂直になる位置で止まるように設計していたが、この設計上ではばねの力が少しでも強い場合、すぐに安全装置が跳ね上がってしまうことが分かっていった。そのため、安全装置 2 号ではばねが地面と垂直な状態よりも少しずれた位置でとまるように設計した。それに合わせて、Selfi 本体の接点部分で後ろに進まなくなるように抑制を加える部分も調整した。また、Selfi 本体との接点を持つ部分が 4 つに増えたので、摩擦パーツもそれに対応して設計し直した。1 つの摩擦パーツには、前と後ろにひとつずつ変動パーツを取り付けそれを二組用意しつなげた形となった。



図 5.24 安全装置 2 号

結果 安全装置 2 号では、ねじの動く範囲を垂直よりも傾きを持つようにしたので少し短いばねを使うことができるようになった。また、ばねが少し縮まった状態になるので安全装置をセットしたときに安定するようになった。ばねの数も 4 つと増やしているためテスト用の安全装置よりも跳ね上がりが良くなった。Selfi 本体の接点も 4 つに増えたことと、地面に接するパーツを左右対称にしたことからバランスが良くなった。こうしたことによって、安全装置が跳ね上がった時 Selfi 本体にぴったりと沿って固定されるようになったので、走行中に邪魔になることもなくなった。また、安全装置 2 号でも後ろへいかないような抑制ができていることが確認できた。図 5.25 に安全装置のセット前、図 5.26 にセット後の様子を示す。

まとめ 安全装置は、強度の問題や取り付けの問題が解決できなかったが安全装置の機構を試作品として完成することができた。この安全装置を使用することで、搭乗の際に後ろへ進まないような抑制を加えることが可能となった。

(※文責: 中田智子)



図 5.25 安全装置セット前



図 5.26 安全装置セット後

Android 端末の Selfi への取り付け

このレンタルサービスでは Android 端末が Selfi を貸し出す際に必要となっている。そのため、Android 端末が Selfi の近くになければならない。Selfi にカゴなどを取り付け、そこに Android 端末を入れておくことも考えたが、それでは走行中の衝撃などで壊れてしまう可能性があった。なので、Selfi 本体に取り付けることで走行中の衝撃をなくすことを考えた。

Selfi 本体に取り付けるにはどのようにすればいいか、最初に浮かんだ方法は Selfi 本体にボルトやナットで直接取り付ける方法だった。Android 端末を覆うケースを作成し、そのケースと Selfi 本体をボルトとナットで固定することによって走行中の衝撃を緩和することができると考えた。しかし、この方法だと Android 端末の画面が固定されてしまい、レンタルの際 ID やパスワードが入力しにくいことが推測できた。サービスでは「使いやすさ」も重要なファクターだと考え、新たに「使いやすさ」を追加することとなった。そこで、Android 端末の角度を調節できるように Selfi 本

Make Use of Personal Mobility for Indoor Movement

体とケースの間に関節を増やすことで使いやすくなると考えた。Selfi 本体と関節, そして Android 端末のケースをボルトとナットで固定する方法で Android 端末を取り付けることとなった。

今回のレンタルサービスで Android 端末を利用するのだが, この端末の電池量では途中で電池切れになってしまう可能性があった。なのでこの Android 端末のバッテリー拡張パックも用意することとなった。これも Android 端末が近くにないと効果がないので, このバッテリー拡張パックを関節部分に取り付けることとなった。

Android 端末のケース Android 端末を覆うケースの作成に取り掛かった。今回利用する Android 端末は Nexus7 で, 採寸した結果, 横 114mm, 縦 200mm, 幅 10mm だった。Nexus7 を覆うケースを作る方法として, アクリル板をレーザーカッターで加工しボルトやナットで固定した。使用したアクリル板は 3mm のものと 5mm のもの, またケースと Nexus7 の間に 5mm の衝撃緩和材を使用した。

今回作る Android 端末のケースには条件があった。ソフト班から Android 端末から音を流すのでその音を阻害しないようなケースにして欲しいとの要望があったので, アクリル板がスピーカの部分に被らないケースを作る必要があった。そして, Android 端末の電源ボタンと音量のボタンを押せるようにすること, またアンドロイド端末が紛失しないような工夫をする必要があった。最後に, Android 端末は縦向きに使われるので縦長のケースを作成した。

最初に DraftSight を利用して CAD データを作成しそれをレーザーカッターで加工その後ボルトとナットで固定した。このときアクリル板を直角で加工してしまうと手を切ってしまうので丸みを帯びさせて加工し, その上で鑢がけをした。背面には関節部分と取り付けるための穴を開け, 右側にボタンが押せるような空間を作り, 表面には鍵を取り付けるための穴を開けた。図 5.27 に示したものが今回作成した Android 端末のケースである。

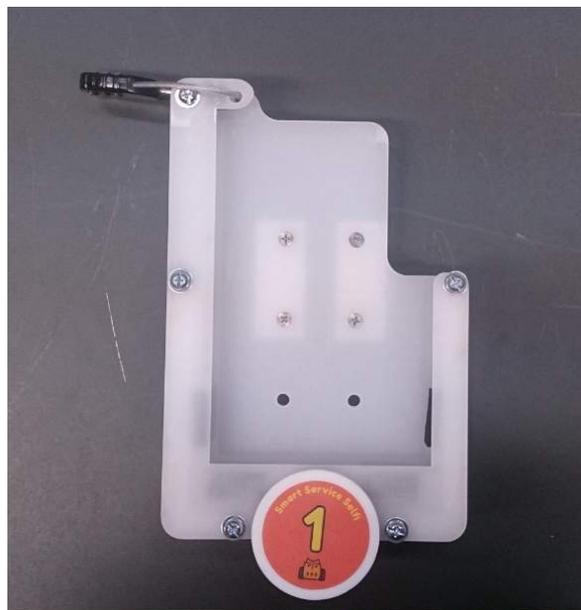


図 5.27 Android 端末のケース

関節部分とバッテリー拡張パックのケース 関節部分は Android 端末が十分使いやすい位置になるような長さを想定して作った。上の部分には Selfi のハンドルユニットの棒にあわせて設計し、Android 端末を取り付けるための穴を調節しながら作った。

バッテリー拡張パックのケースについて、作るために必要なバッテリー拡張パックの採寸をした。その結果、横 100mm、縦 42mm、幅 23mm であった。このバッテリー拡張パックを作る方法として、アクリル板をレーザーカッターで加工し、蝶番をつけたあとにアクリル板用の接着剤で固定した。使用したアクリル板は 3mm のもので、蝶番は百円均一の店舗で購入したものである。

今回作るバッテリー拡張パックのケースは底の部分からコードを伸ばし、Android 端末に接続する必要があった。また、このバッテリー拡張パックを取り出したり、紛失しないような工夫が必要であった。

バッテリー拡張パックは Android 端末のケースと同じ作り方をしている、レーザーカッターを使用した。底の部分の穴の大きさは横 35mm、縦 18mm のものである。

作成した関節部分とバッテリー拡張パックをアクリル板用の接着剤で固定した。図 5.28 のものが今回作成した関節部分とバッテリー拡張パックである。



図 5.28 関節部分とバッテリー拡張パック

最後に、これらのパーツを取り付けて不具合がないかを確認した。不具合は特に見られなかったので衝撃緩和材を使用して完成とした。図 5.29 は実際に組み合わせたときの裏側のものである。

(※文責: 橋本隆義)

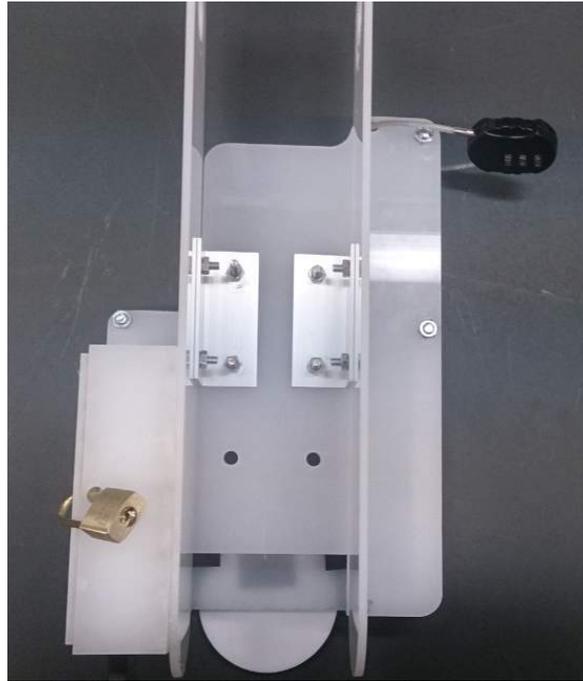


図 5.29 作成したケースの裏側

Arduino を搭載するための回路作成

Arduino で製作した機能を実際に Selfi に取り付けるため、回路を作成した。その回路図を図 5.30 に示す。Arduino の電源は Selfi のスイッチ部分からつなぐことで、Arduino は Selfi の起動と同時に起動するようにした。また、図 5.31 に実際に Selfi に Arduino を搭載したものを示す。

(※文責: 山本一希)

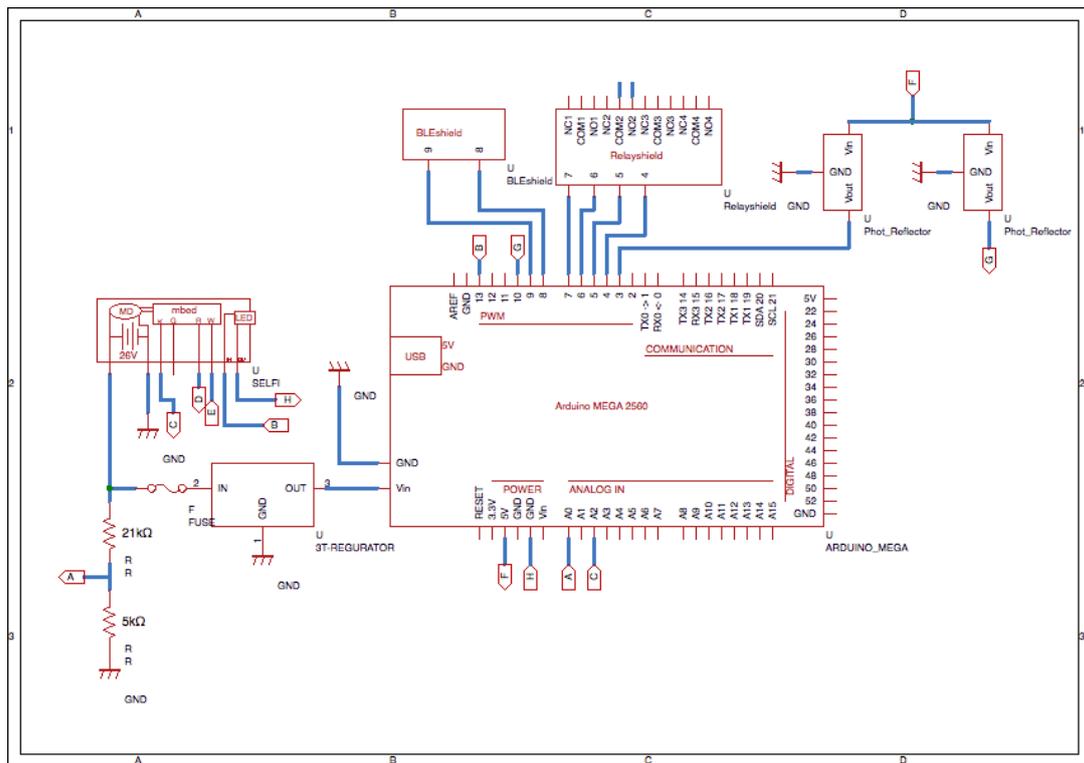


図 5.30 回路図

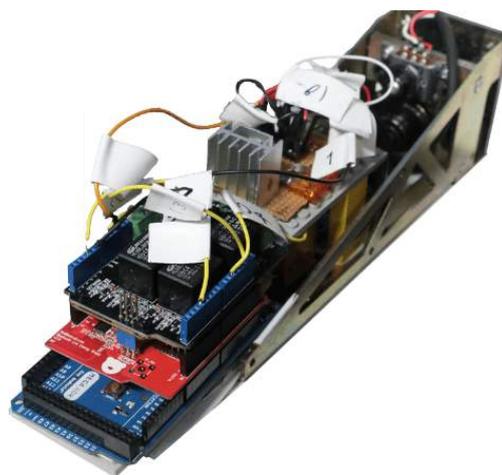


図 5.31 Selfi に搭載した Arduino

5.3 PR 活動の課題解決プロセス

以下に、4.3 節で述べた課題解決のために、実際に行った PR 活動の課題解決のプロセスを示す。

- PV 動画の製作.
- イメージキャラクタの製作.

これらの各項目について、以下に詳細を述べる。

PV 動画の製作

どのような PV 動画を作るか検討した結果、Selfi を宣伝する動画 (図 5.32) とレンタルサービスの流れを説明する動画 (図 5.33), Selfi の乗り方を説明する動画 (図 5.34) の 3 つを作ることとなった。動画の製作の際には iMovie というアプリの映画予告編というテンプレートを使用した。試験運用や最終発表などでスクリーンに宣伝広告として映した。



図 5.32 SSS



図 5.33 SSSmanual



図 5.34 Selfi の乗り方

(※文責: 山本一希)

イメージキャラクターの製作

本プロジェクトの目標として「不特定多数の方に Selfi のレンタルサービスを使用してもらう」というのがある。実際に運用していく際に、多くの人に周知され、かつ、サービスの知名度を向上するための方法としてイメージキャラクターを製作した(図 5.35)。このイメージキャラクターは Selfi とフクロウの二つの要素からなっている。フクロウという動物をモチーフにしている理由として、まず、フクロウに漢字をあてると「不苦労」となる、さらに、Selfi に乗って苦労することなく楽に移動ができるという意味を込めているからである。また、Selfi を正面から見ると、フクロウの羽の部分が Selfi のタイヤ、フクロウの眉毛の部分が Selfi のハンドルに似ていると感じたからである。フクロウのお腹にある3つの「S」は「Smart Service Selfi」の頭文字をとったものである。「Smart Service Selfi」とは、本プロジェクトのサービスの周知させるためのキャッチコピーとして、イメージキャラクターを同時に考案したものである。



図 5.35 イメージキャラクター

今回のイメージキャラクターの製作は前期の議論の中で、案として上がっていなかった。後期の活動の中でハードウェア班の活動が個人単位となり、仕事の割り振りに差ができてしまった。そこで、残りのメンバで議論し、本プロジェクトで欠けていることはなにか等を話し合った。その中で、本プロジェクトはサービスを運用するという上で、システムの開発に力を注いでいたため、サービスを多くの人に認知してもらう活動をしていないという問題があった。そこで、PR のやり方について議論し、イメージキャラクターの製作に至った。

イメージキャラクターの利用方針として、議論の中では LINE のスタンプ・タオル・ワッペン・缶バッジ等、グッズによるサービスの周知向上を考えた。しかし、それでは、本来の目的から大きく離れてしまうため今回は以下のものをイメージキャラクターを用いて制作した。

ナンバープレート

本プロジェクトは、今年度作成した Selfi 以外にも前年のプロジェクトにおいて作成した Selfi もあり、合計 3 台所有している。今回のサービスを展開していく中で複数の Selfi を稼働させることが念頭にあったため、個々の Selfi を識別する必要があった。そこで、製作したイメージキャラクターを用いて 3 種類の Selfi のナンバープレートを製作した (図 5.36)。

ナンバープレートは、Androido ケース下部に装着した (図 5.37)。ナンバープレートに用いたフクロウの配色は「Adobe Kuler」を用いて、アプリケーションの配色でもあるオレンジ系統の配色を中心として決めた。図 5.38 に示したような多くの種類のなかから 3 種類をプロジェクトメンバーに決めてもらった。



図 5.36 ナンバープレート



図 5.37 装着したナンバープレート



図 5.38 配色デザイン

看板

後期の活動として、サービスの試験運用を行った。その際に、試験運用を行う場所をわかりやすくするためにイメージキャラクターを用いて、看板を製作した(図 5.39)。素材は MDF 材とアクリル板を使用した。MDF 材はそのまま加工せず、アクリル板をレーザーカッターで加工した。アクリル板の色が白であったため、アクリル絵の具を使用し、レーザーカッターで加工したものに色を塗った。加工したアクリル板に色を塗ることができたが、絵の具が乾燥すると色が剥がれてしまう問題があった。



図 5.39 看板

チームパーカー

プロジェクト内の士気向上，団結力の強化のためにイメージキャラクターを用いたチームパーカーを製作した(図 5.40)。チームパーカー用にイメージキャラクターにアレンジを加えた。担当教員全員がメガネをかけていたので，メガネの位置によってそれぞれの担当教員をイメージする形としたデザインをパーカーの胸のところに印刷した(図 5.41)。バックにはタイヤをイメージしたデザインを用い，その上にプロジェクトメンバと担当教員の名前をプリントすることとした。



図 5.40 チームパーカー (左：表，右：裏)



図 5.41 アレンジを加えたイメージキャラクター

まとめ

今回のイメージキャラクター作成では、フクロウ以外の図案もあった。本大学の教室名には別名があり、図書館が「灯台」、講堂が「クジラ」という海にまつわる単語の名前も持っている。そのことから、海に関係する生き物のイメージキャラクターも考案した。今回は採用されなかったが、Illustrator の勉強をすることができた。

以上の活動を通して、11月26日から11月28日の3日間の試験運用は本大学の生徒・教職員の目に止まることができたので、サービスを体験する人を多く集めることができた。

(※文責: 鈴木絢子)

第 6 章 Selfi の試乗とサービスの試験運用

6.1 試乗での活動

6.1.1 オープンキャンパス

2014 年 8 月 2 日公立ほこだて未来大学のオープンキャンパスの学科出展に参加した (図 6.1)。ここでは Selfi を試乗してもらう場とし、一般の方に対してアンケートを実施した。

初めは 2 台の Selfi を稼働させ、1 台を予備として準備していたが中盤からは Selfi の故障が目立ち、多くの人に乘ってもらうことは困難となった。故障の原因はほぼ制御ボタンの接触不良であり、初心者はボタンを強く押してしまうために壊れやすくなってしまうことが分かった。これにより制御ボタンにより Selfi の制御を入れるのではなく、マイコンにより本体の電源が入った時点で Selfi の制御が入る状態にするというように改良することを考えた。

さらに Selfi のタイヤの空気が抜けてしまい、交換を試みたがタイヤのチューブに穴が開いていたため交換することができなかった。これによりタイヤのチューブをより素材が丈夫であるものに交換する必要があるという問題点も発見することができた。

アンケートの結果としては、「面白かった.」「楽しかった.」などの感想が多く、さらに近場の移動や、大型施設での移動に利用したいなどサービスとしての需要があることが分かった。一方で、「慣れるまではバランスを取ることが難しい.」という意見が多く、初心者でも安定して走行できるような工夫も必要であることが分かった。

オープンキャンパスで試乗を実施したことにより、制御、バランスを取る部分についての改良が必要であることが分かり、サービスを運用していく上での問題点を見るいい機会となった。



図 6.1 オープンキャンパスでの試乗の様子

(※文責: 菊地亜美)

6.1.2 HAKODATE Developer Conference

2014年10月25日、函館市中央図書館にてはこだてディベロッパーカンファレンス2014が開催された。IT技術の講演会などを目的としたイベントで今回のプロジェクトでは常時開設ブースとして出展をした(図6.2)。

図書館の一角という狭いスペースだったため、あまり試乗の際に激しい動きはできなかったが会場関係者も含め多くの方に乗ってもらえた。中には小学生もおり、幅広い層に乗って頂けた。また今回の試乗は安全装置やAndroid端末等を搭載せず素のSelfiを使用して実施した。試乗してくれた方の感想は「最初乗るときが怖かった」、「慣れるとおもしろい」などSelfiへの感想が多く肝心のサービスへの感想は得られなかった。



図 6.2 HAKODATE Developer Conference での試乗の様子

(※文責: 金井貴浩)

6.2 試験運用の実施

6.2.1 試験運用の概要

目的

本プロジェクトで製作したレンタルサービスを実際に稼働させることによって、サービスが実用可能であるかを検証すると共に、単純なSelfiの試乗とは異なる、レンタルサービスを体験したユーザの高いレベルのフィードバックが得ることが可能となる。ユーザ視点、開発者視点の異なる観点からの評価を受けることで、現段階でのレンタルサービスの適切な評価を行い、サービスの問題点や、改善すべき点を明らかにする。

日時と場所

2014年11月26日から2014年11月28日にかけて、公立はこだて未来大学三階エレクトロニクス工房前にて試験運用を行った(図6.3)。日時の詳細は表6.1の通り。

表 6.1 試験運用の日時の詳細

日付	開始時刻	終了時刻
11月26日	15:00	18:00
11月27日	11:00	15:00
11月28日	15:00	18:00



図 6.3 試験運用の様子

使用機材

以下の改造を施した Selfi 一台。

- Nexus7 用のケースの設置。
- Nexus7 のバッテリー拡張パック用ケースの設置。
- Selfi の制御の切り替え，速度の計測，バッテリー残量の計測，Nexus7 との通信を行う Arduino の設置。
- 速度計測用フォトリフレクタの設置。
- Selfi の番号を記したプレートの設置。

なお，Selfi の安全装置は開発途中であったため装備していない。

- サービス用サーバ機。
- LAN 構築用有線 LAN ルータ・無線 LAN ルータ・LAN ケーブル。
- 本プロジェクトの制作したアプリケーション Selfit をインストールした Nexus7。

環境

全ての通信は LAN 内もしくは、Bluetooth によって Arduino・Nexus7 間で行われ、インターネットを介した通信は一切行っていない。図 6.4 に試験運用時のネットワーク構成を示す。

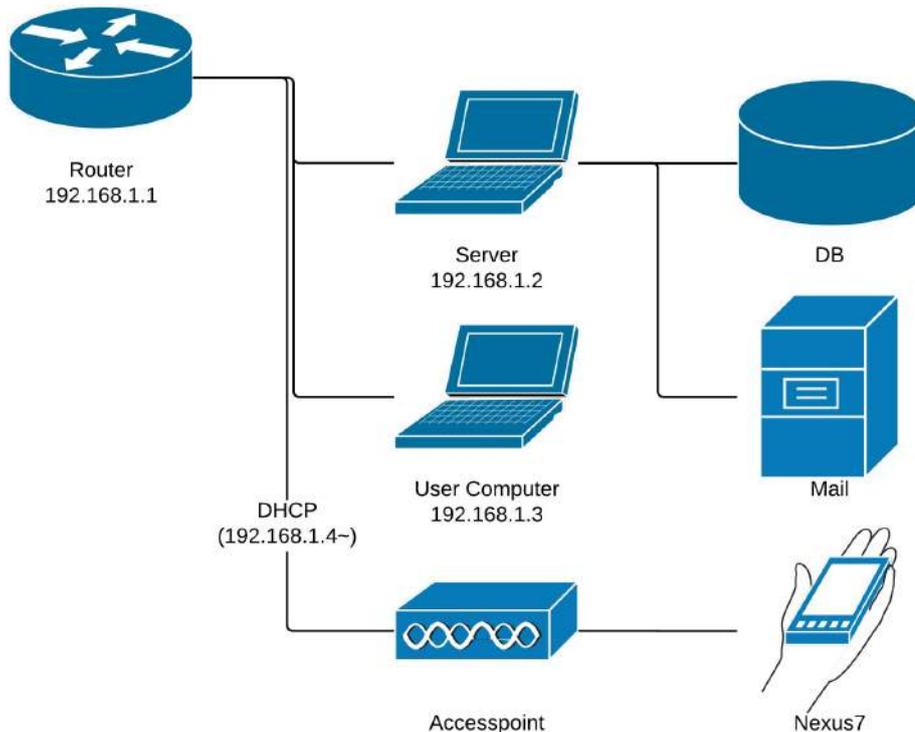


図 6.4 試験運用時のネットワーク構成

手続き

今回の試験運用では、限られた時間内で、レンタルサービスの一連の流れをユーザに体験してもらうことを重視した。ユーザがマニュアルなどを読んでからサービスの利用を開始するというような、利用法について何の知識も無い状態だと、使用するまでに多くの時間がかかると考えたため、貸出から返却までをユーザに全て任せるのではなく、プロジェクトのメンバが、ユーザにサービスの利用の手順や Selfi の乗り方の説明を行った。ユーザが Selfi を返却した後は、レンタルサービスの利用に関するアンケート調査を行った。

(※文責: 吉田豊)

6.2.2 試験運用の評価

試験運用で行ったアンケートの結果を図 6.5 と図 6.6 に示す。

まず、図 6.5 に示した「サービスをまた利用したいですか?」という質問に対する回答では、9割以上の回答者にまた使いたいと回答してもらうことができた。この結果から、Selfi のレンタルサービスは需要があるということが確認できた。

Make Use of Personal Mobility for Indoor Movement

一方で、図 6.6 に示した「危険を感じましたか?」という質問に対する回答では、約 4 割の回答者が危険を感じたという結果であった。この結果から、安全面での対策の必要性を再確認し、安全装置の重要性を裏付ける結果が得られたと考える。

また、アンケートに記入してもらった意見からいくつか課題が得られた。以下にそれらの課題と解決案について述べる。

課題：初めて Selfi に乗る際に、乗り方が分からない

解決案： 今年度はマニュアルを作成したが、マニュアルを読むだけでは不十分であると考えられる。そこで、音声案内で乗り方のレクチャーを行うなど、より分かりやすい方法でユーザに乗り方を伝える必要があると考える。

課題：手を離すと Selfi が倒れてしまう

解決案： Selfi が倒れないようにするスタンドを作るための図面は作成済みであるが、Selfi 自体の重量から素材の耐久性が問題となっていた。よって、Selfi の重さに耐えられる素材を見つけ、それを用いてスタンドを作成することで解決できると考える。

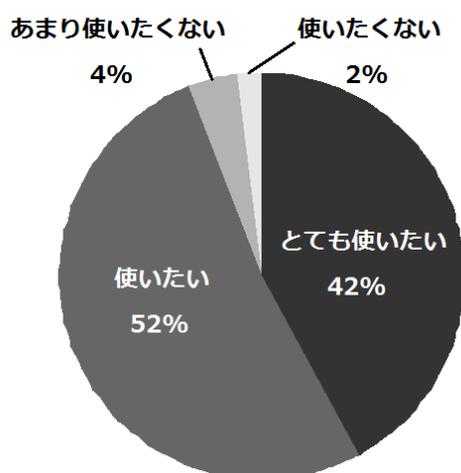
課題：Android 端末の位置が低いため、使いにくい

解決案： Android 端末の位置を Selfi の高い位置に取り付けることで使いやすさを向上させることができると考えられる。しかし、Selfi の重心位置が高くなることでバランスが悪くなり、Selfi が走行中に不安定になる可能性も考えられる。したがって、Android 端末を取り付ける位置は慎重に検討する必要があると考える。

課題：Android 端末で速度を確認しようとする、目線が下に行き危険である

解決案： 視覚以外の手段で速度を報知することで解決が可能であると考えられる。具体的には、速度に応じて Selfi の接近報知音の音量を変えるなど、聴覚に働きかける方法が最も効果的であると考えられる。

Q. サービスをまた使いたいですか?



Q. 危険を感じましたか?

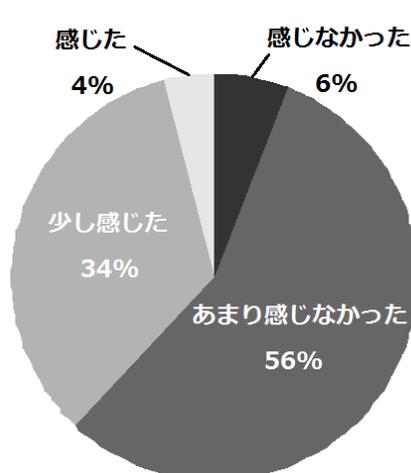


図 6.5 試験運用のアンケート結果 1(回答者:61 人) 図 6.6 試験運用のアンケート結果 2(回答者:61 人)

(※文責: 水澤吉博)

第 7 章 まとめ

7.1 プロジェクトの成果

今年度の目標である、公立はこだて未来大学でのレンタルサービスの実現については、開発したシステムの試験運用を行うことができ、開発したシステムを用いてサービスを運用することが十分可能であることを確認することができた。よって、今年度の目標は概ね達成できたと考える。しかし、初心者に対する対策が不十分であるなどの問題点も見つかったため、それらの問題点を今後改善していくことで公立はこだて未来大学での実運用を実現できると考える。

また、本プロジェクトの目的についても、今年度開発したシステムを用いて、空港やショッピングモールなどの広い屋内でのレンタルサービスに応用できると考える。具体的には、サーバをインターネット環境で稼働させ、メールアドレスやユーザの個人情報の登録を行う仕組みを作ることで応用可能であると考え。このシステムを用いたレンタルサービスを運用することによってパーソナルモビリティを不特定多数の人に利用してもらうことができ、普及させることができると考える。

以上のことから、今年度のプロジェクトでは目的の実現に役立つ成果が得ることができた。

(※文責: 西村祐輝)

7.2 システムの各構成要素ごとの成果

7.2.1 アプリケーション

7.2.1.1 Android アプリ

レンタルサービスに使う Android アプリケーションの開発を行った。この Android アプリには次のような機能を搭載した。

- パスワード、ID、Selfi の状態などログイン情報をサーバと交換する。
- Android アプリケーションにユーザがログインした際、Bluetooth を通じて Selfi に制御を入れる。
- レンタルサービスに必要なマニュアルなどの情報をユーザに提示する。

また、Android 端末は Nexus7(2013) を使用し、Selfi 本体に設置した。以上の Android アプリを搭載した Selfi を図 7.1 に示す。

今回のアプリケーションにはデータサーバとの通信に JDBC を使用した。この方法は Android アプリケーションの開発においては普通ではない。通常は HTTP サーバなどを經由してデータベースサーバに接続して行うのが普通だ。しかし今回は JDBC という MySQL に Java から接続するものを使用した。この方法はあまり一般的ではなく JDBC が Android に使用している ART、もしくは Dalvik という Java 仮想マシン上では動くという保障はない。実際に今回の開発では JDBC のバージョンの違いによって接続できる、できない等の問題が発生した。よって今後のサーバとの通信では HTTP サーバ等を使い実装をしていくほうが良いと思われる。またサーバ側に問



図 7.1 Android アプリを搭載した Selfi

い合わせ、持ってきた ID とパスワードだがこれは平文で保存しておりセキュリティ上大きな問題と考える。暗号化して保存するか、ユーザから入力された ID とパスワードを認証するごとに破棄し、次に認証するときはサーバに取りに行くといったローカルには保存しないなどの方法を取ったほうが良いと思われる。今回の Android アプリケーション開発では速度に合わせて接近報知音を流す、といったことが実装できなかったためこの機能を次回の Android アプリケーション開発では実装する必要がある。

しかし今回の試験運用では十分な働きをしたのでセキュリティ面を主に改善をすれば十分実用レベルのものと考えている。

(※文責: 金井貴浩)

7.2.1.2 音

今回のプロジェクト学習では公立はこだて未来大学の 3 階モールにて Selfi のレンタルサービスを運用するというので 3 階のモールに合わせた Selfi の接近報知音を検討した。具体的には 3 階のモールにおける環境音が 56dB なのでそれに合わせた 66 d B の接近報知音を検討した。接近報知音として騒がしい環境の中でも聞こえて Selfi のことを認識できる、なおかつ不快な音ではないことを目標とした。

音の印象実験などを行った結果、329Hz の「ビー」という単調な音に決定した (図 7.2)。

今回の接近報知音の検討は音階だけであり、その他の音の要素、音の大きさなどを検討していく必要がある。また最終的な音の印象実験については標本数の人数があまりにも少ないため結果の偏り等が十分考えられる。

また接近報知音に適する音は対象となるものが近づいてくるとわかる音がベストである。今回はビーという単調な音なの Selfi が近づいてくるとはわかりづらい。よって次回に検討する際は Selfi のモーター音などの音が最適と考える。

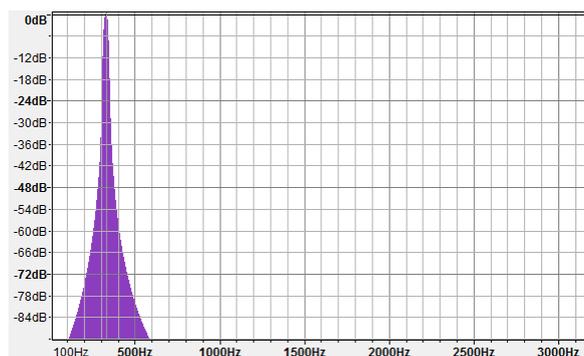


図 7.2 周波数解析

(※文責: 金井貴浩)

7.2.2 サーバ

サーバでは主に、データベースサーバ、メールサーバの構築、自動的に実行されるプログラムの作成を行った。データベースサーバでは、MySQL を使いレンタル状況や Selfi の状態、ユーザの情報をデータベースで管理している。管理することで、Selfi が利用可能かどうか判断できる。メールサーバでは、ユーザとのメールの送受信を行う。ユーザからのメール受信で利用申請がされ、ユーザへのメール送信でパスワードや申請完了等が送られる。プログラムはユーザからの申請メールを受信した際に実行される。実行される動作は、データベースにアクセスし Selfi が利用可能かどうか判断する、状況に応じてメールを送信する、データベースを更新する等である。また、ローカル環境での効果は見込まれないが、メールの送受信、データベースへの通信を暗号化した。

(※文責: 水澤吉博)

7.2.3 マイコン

本プロジェクトの活動によって、当初から考えていた機能を確認し、Selfi への設置することができた。Arduino への電源供給の確保は、レギュレータとヒートシンクを使うことによって Selfi の電源をバッテリーから確保した。Selfi のロック機能はリレースイッチによって容易にロックすることを可能とした。バッテリー残量は分圧によって誤差 1% の計測が可能とした。Bluetooth による Android 端末との通信機能は BLE シールドを使うことによって可能とし、それぞれの機能で得た情報を Android 端末に送信することを可能とした。また、今後の課題として、今の電源供給システムと Selfi のバッテリーでは、充電を行いながら Arduino を使用することはできないことがあげられる。

(※文責: 山本一希)

7.2.4 ハードウェア

ハードウェアの成果は、Selfi の組み立てと安全装置、Android ケースである。まず Selfi の組み立てに関してだが、Selfi 自身の仕組みを理解するためにプロジェクト活動内で組み立てを行った。Selfi は大きく分けてドライブユニット、メインフレーム、バッテリーホルダ、ハンドルユニット、基板の 5 個の要素からできている。Selfi の組み立ては外側だけではなく、制御に関するキャリブレーションの調整も行った。また、ある程度外側の組み立てが終わってから外側と制御の細かいところを調整した。次に安全装置についてだが、今回は素材の強度から、試作品までの完成が目標としていた。作成にはレーザーカッターと CAD データ、アクリル板を使用した。ばねの力を利用し、走行中には邪魔にならずなおかつ搭乗の際のぐらつきを減少させるための装置を作成した。強度の問題から、実装することはできないが、Selfi 本体の下部分に安全装置を設置することが可能であることを確認できた。また、機構として今回作成した安全装置の構造が実装につながると考えられる。最後に Android 端末のケースについてだが、サービスに必要となる Android 端末を Selfi 自体に取り付けるためのケースを作成した。作成にはいくつか条件があり、スピーカーに被らないこと、電源ボタンを押せること、音量調節ボタンを押せること、Android 端末が紛失しないように工夫することがあげられた。また、Android 端末ケースのほかにも Android 端末のバッテリー拡張パックも作成しケースと一緒に取り付けた。バッテリー拡張パックにも紛失を防ぐため施錠できるよう鍵穴を付けた。

(※文責: 中田智子)

7.2.5 PR 活動

PR 活動として、まず、本プロジェクトに情報デザインコースの学生がいなかったことから、イメージキャラクターの製作において多くのことを学ぶ必要があった。イメージキャラクターとロゴの違いはどこにあるのか、イメージキャラクターの役割、イメージキャラクターによる影響など、一から勉強することが多々あった。また、チームの団結力を高めるためにチームパーカーを製作したことで、プロジェクト内の士気向上を図ることができた。試験運用の際に製作した、宣伝動画、看板により、本学内の生徒・教職員の意識をサービス会場に向けることができた。ナンバープレートは、もう少しわかりやすい位置に配置する必要があることも試験運用の際に実感した。全体を通して、PR 活動を行うことでよりサービスを展開しやすいことが今回わかった。

(※文責: 鈴木絢子)

7.3 発表の評価

7.3.1 中間発表会

2014 年 7 月 11 日に行われた中間発表の様子を図 7.3 に示す。中間発表では、発表評価シートを配布することで、プロジェクトの活動について効果的な発表が行なわれているかを問う発表技術、発表内容について来場者に評価してもらった。評価の内容として、「実演があったので分かりやすかった」、「ぜひ実現して欲しい」や「実際に Selfi を体験出来てよかった」などの評価があった。しかし、発表技術の基本的な意見としては「声量が少なかった」や「スライドをもっと工夫した方が

Make Use of Personal Mobility for Indoor Movement

いい」など発表の技術や演出はまだ不十分であるとの指摘を受けた。以上のことを踏まえて最終発表ではスライドの工夫してわかりやすくするなど発表技術で参考にした。

発表内容は、「本当に Selfi を学内でサービスとして繰り返し使ってもらえるのか」や「なぜ Selfi をユーザに利用してほしいのか、またその利点も欲しかった」など、Selfi をサービスとして利用してもらうための内容が薄かったことが分かった。したがって、最終発表の時は Selfi のサービスについて聴講者が理解出来るような内容にしておくため分かりやすい説明をした。また、警告音についての意見が賛否両論であったこと、安全性に対するハード的な内容の意見が多数あったので、音やハード面でのアンケートを取るなど、後期の活動ではこれらを取り入れていった。



図 7.3 中間発表の様子

(※文責: 高村芳明)

7.3.2 最終発表会

2014年12月12日に行われた最終発表の様子を図7.4に示す。最終発表では発表評価シートにより、プロジェクトの活動についての評価を得た。評価シートの内容については、プロジェクトの内容について効果的な発表が行われているかを評価する発表技術について、さらにプロジェクトの目標設定と計画は十分なものであるかを評価する発表内容についての2項目で評価してもらった。

発表技術については「流れが途切れない構成だったので聞きやすかった。」や「動画を使ったプレゼンは分かりやすかった。」など発表の流れについては良い評価をもらった。さらに動画やスライド、ポスターについても「良く出来ていた。」というコメントがほとんどであった。しかし声量について書かれていることが多く、「発表が聞こえなかった。」というコメントも目立っていたため、発表者の選出について考える必要があることが分かった。

発表内容についてはサービス全体の評価として「システムとして成立している。」や「よく考えていると思う。」などサービスとして利用していきたいという意見が多かった。「ターゲットが気になった。」という意見も多く、発表の中にうまく取り入れることが出来ていなかったことが分かった。

全体としては好意的な意見が多く、プロジェクト全体として成功したという評価をもらうことができたと言える。

(※文責: 菊地亜美)



図 7.4 最終発表の様子

7.4 今後の課題と展望

今年の本プロジェクトの目標である、公立はこだて未来大学内で大学関係者を対象とした Selfi のレンタルサービスを構築することを達成するために、前期の活動として目標設定から、レンタルサービスの流れや、Selfi の製作、サービスに必要な前提知識などの学習や実験を行った。後期では前期の活動をもとにシステム完成に向けて、アプリの作成やサーバの設置、マイコンの組み込み、安全対策を行った。そしてシステム完成後、サービスの試験運用を行いサービスの有用性と今後の課題を見出すことが出来た。

今後の課題として、安全性、初心者対策、サーバのインターネット接続、サーバの充電中 Selfi のバッテリー残量の把握の大きく4つが挙げられる。

ハード面では安全対策がまだ不十分であるという現状が挙げられる。まず、現状では Selfi に自立性がないため、手を放すと倒れてしまう。そのためのスタンドを兼ねた安全装置の試作品は完成しているが、素材の強度に問題があるため実装には至っていない。また、視覚以外での速度報知の必要性があり、これは周知音を速度に応じて音量を変えて流すなどの方法を検討している。そして、Selfi に取り付けられた Android 端末の設置位置が低く、アプリが使いにくいという点が挙げられるが、端末の位置を高くしてしまうと Selfi の重心に問題が出てしまい安全性に欠けるため検討が必要となる。この3つのことから、安全対策にまだ不安がありこれからも対策を講じていく必要性がある。

アプリ面では、初心者対策をより重要視して行っていく必要がある。アプリ内にマニュアルを内蔵してあったが、試験運用では利用されることはなく、プロジェクトメンバが乗り方について説明していたため、アプリのマニュアルは未だ初心者が読んで理解するにはほど遠いものがある。そのため、マニュアルに関しては、改良を加える必要がある。

サーバ面では、WiFi を通してサーバと Android 端末間の通信を行っており、WiFi のつながる範囲でしか通信を行うことが出来ず、試験運用の際には Selfi にのって遠くに移動する手段としての利用が出来なかった。そこで、サーバをインターネットに繋げて、Android 端末との通信を行う必要性が出てきた。もう一つは、サーバが Selfi のバッテリー残量管理をうまく行えないという点である。最初の構想では、Selfi のバッテリー残量をマイコンから Android 端末経由でサーバに送信するというものであった。しかし、Selfi のバッテリーの残量を送信するためには、マイコンを使用するために Selfi の電源を入れなければならない。そこで問題になったのが、Selfi の電源をいれ

Make Use of Personal Mobility for Indoor Movement

てしまうと同時に充電ができないということだ。Selfi を充電する際には、電源を切った状態にしておかなければならないため、現在の状況では Selfi のバッテリー残量をサーバ側で完全に把握することが出来ない。そのため、今後はサーバ側で充電中も Selfi のバッテリー残量を管理できるようなシステムをもう一度検討しなおす必要がある。

(※文責: 芹澤雪花)

参考文献

- [1] 株式会社エフ・アイ・ティー, ”FIT-Robots”,
<https://sites.google.com/site/fitrobots/home> (2014/12/22 アクセス)
- [2] セントレア, ”セグウェイガイドツアー”,
<http://www.centrair.jp/enjoy/visit/segway/> (2014/12/22 アクセス)
- [3] ASCII.jp, ”中部国際空港での “セグウェイ” を体験してみないか?”,
<http://ascii.jp/elem/000/000/405/405045/> (2014/12/22 アクセス)
- [4] セグウェイジャパン, ”Segway とは?”,
<http://www.segway-japan.net/technology/segway/> (2014/12/22 アクセス)
- [5] セグウェイスマイル, ”セントレア 【中部国際空港】 セグウェイ・ガイドツアー”,
<http://www.centrair.jp/enjoy/visit/segway/> (2014/12/22 アクセス)
- [6] 国土交通省, ”ハイブリッド車等の静音性に関する対策のガイドライン”,
<http://www.mlit.go.jp/common/000057372.pdf> (2014/12/22 アクセス)
- [7] 岩宮眞一郎, サイン音の科学 -メッセージを伝える音のデザイン音-, pp.67-75, コロナ社, 2012.