

コンピュータ理工学特別研究報告書

題目

Raspberry Pi と顔認識カメラを使った
自動入退室システムの開発

学生証番号 1545150

氏名 花木 泰秀

提出日 平成 31 年 1 月 22 日

指導教員 蚊野 浩

京都産業大学
コンピュータ理工学部

要約

カードキーを使った電子錠はカードキーの置き忘れによる締め出しや、開錠／施錠の煩わしさを回避するためにドアを開けっ放しにすることでセキュリティが低下するという問題がある．本研究では、スマートロック技術を参考に、既存の入退室システムにアドオンして動作する顔認識開錠システムを、Raspberry Pi と顔認証カメラなどのコンポーネントを用いて開発した．

開発したシステムは、顔認識を行う廊下側端末と、サムターンを回転させ開錠する室内側端末から構成され、この2つの端末が無線で通信する．廊下側端末は顔画像を認識する機能を持つカメラで個人認証を行う．認証に成功した場合、個人の ID を室内側端末に送信する．室内側端末はサムターンを回転させる機構部とディスプレイを備えている．廊下側端末からの開錠指示か、ディスプレイに表示される GUI の操作による開錠指示が発生した場合、サムターンを回転させて開錠する．サムターンを回転させるメカニズムは3Dプリンタで自作した機構部とサーボモータで構成される．室内側端末は、所属メンバーの在室状態を管理しており、その情報を使った GUI がディスプレイに表示される．廊下側と室内側の端末間の通信には無線モジュールを利用した．開発したソフトウェアは入室動作と退室動作の2つに分かれている．これらの動作毎に利用したメンバー名、在室状態、日付、時間の4つの情報を持つログを生成し、これらを管理・確認できるようにしている．

開発したシステムを動作させて機能を検証した．入室の場合、廊下側端末に顔を映してから約2秒で開錠した．退室する場合、ディスプレイ上の退室ボタンをクリックすることで開錠した．いずれの場合もログが正しく残っており、全体として正常に動作した．また、カードキー使わずに入室できることで利便性が高まったことが検証できた．課題として、システムの動作状態が不明で、利用者に戸惑いを感じさせることがある．記録したログの確認が容易でないことや、廊下側端末をモバイルバッテリーで動作させた場合、連続使用時間が半日程度と短いことも問題である．

以上、正しく動作するシステムの開発と実用性の検証ができた．しかし、機能試作品として位置付けられるもので、製品化には、省電力化や小型化、ローコスト化などの課題もある．

目次

1 章 序論	・ ・ ・ 1
2 章 システムの概要と主なコンポーネント	・ ・ ・ 2
2.1 システムの全体像	・ ・ ・ 2
2.2 廊下側端末の構成	・ ・ ・ 3
2.3 室内側端末の構成	・ ・ ・ 4
2.4 サムターンを開錠する機構部	・ ・ ・ 6
3 章 自動入退室システムのソフトウェア	・ ・ ・ 10
3.1 廊下側端末	・ ・ ・ 10
3.2 室内側端末	・ ・ ・ 13
3.3 システム全体の動作	・ ・ ・ 16
4 章 自動入退室システムの動作検証と考察	・ ・ ・ 17
4.1 性能検証	・ ・ ・ 17
4.2 考察	・ ・ ・ 20
5 章 結論	・ ・ ・ 21
参考文献	・ ・ ・ 22
謝辞	・ ・ ・ 22
付録 1 本研究で開発したファイルの構成	・ ・ ・ 23
付録 2 ブート時に指定のプログラムを実行する方法	・ ・ ・ 26
付録 3 使用上の注意点	・ ・ ・ 28

1章 序論

鍵を電氣的に開閉する電子錠が広く使われている．一般家庭やオフィスで使う電子錠は，入室する場合の開錠はカードキーやリモコンで行い，退出する場合の開錠はサムターンを手回しすることが一般的である．ここで，サムターンとはドアの内側についている，鍵を手回しするためのノブである．

最近では，電子錠を高機能化したスマートロックと呼ばれる商品がある．これは，スマホのアプリなどで開錠・施錠ができるようになっている電子錠システムのことである．つまり，スマートロックはネット化された電子錠である．スマートロックの中に，既存のサムターン付き電子錠にアドオンして利用することができる商品がある（例えば Akerun や Qrio Lock など）．アドオン式のスマートロックは安価で導入が容易というメリットがある．また，スマートロックはネットワークに接続しているため，入退室の状況を，ネットワークを介して管理することが可能である．

コンピュータ理工学部がある 14 号館は，学生証をカードキーとして使う電子錠を導入しており，研究室をはじめ様々な施設に出入りできるようになっている．カードキーやリモコンは，通常の鍵に比べると開閉が便利である．しかし，これらのデバイスを携帯する必要がある，トイレ等で部屋を出たときにカードキーを忘れてしまうと締め出されるといった状態になることもある．カードキー忘れによる締め出しを避けるために，マグネット等を使用して自由に出入り可能な状態にしているケースが見られる．これによりセキュリティが低下し，過去に問題が発生した事例もある．このようなことを防ぐ方法として，電子錠にアドオンするシステムで，生体情報で解錠できるシステムを検討した．

生体認証に用いる人体情報には，指紋・顔・静脈・虹彩・網膜など，様々なものがある．今回は，比較的容易に導入することができる顔認識カメラを使った．また，システム全体を制御する組み込みシステムに Raspberry Pi を使い，室外側で顔認証を行う端末と，室内側でサムターンを開錠する端末に分けて実装した．2 台の Raspberry Pi を使った端末の間は，近距離無線で通信させる．サムターンを開錠する機構部は，サーボモータとこれをサムターンに接続するシリンダ状の部品から構成される．3D プリンタを使って，この機構部を製作した．

以下，2 章でシステムの概要と主なコンポーネントについて述べる．3 章で自動入退室システムの詳細について述べる．4 章で動作検証と考察を行い，5 章で結論を述べる．

2章 システムの概要と主なコンポーネント

この章では、自動入退室システムの全体を把握できるように、システムの概要、システムを構成する端末、端末の主要部品、およびシステムの動作シーケンスの説明を行う。

2.1 システムの全体像

図 2.1 を用いてシステムの全体像を説明する。開発するシステムは、廊下側の端末で入室者の顔を認識し、自動的に扉を開錠する。退出する場合には、室内側の端末にタッチすることで自動的に開錠する。廊下側端末の主なコンポーネントは Raspberry Pi Zero W（以下では Zero W）と顔認識カメラ、無線通信モジュールである。室内側端末の主なコンポーネントは Raspberry Pi 2 Model B（以下では Pi2）とタッチスクリーン、無線通信モジュールである。また、自動的に解錠するために、サムターンをサーボモータで回転させる仕組みになっており、これを室内側の端末から制御する。

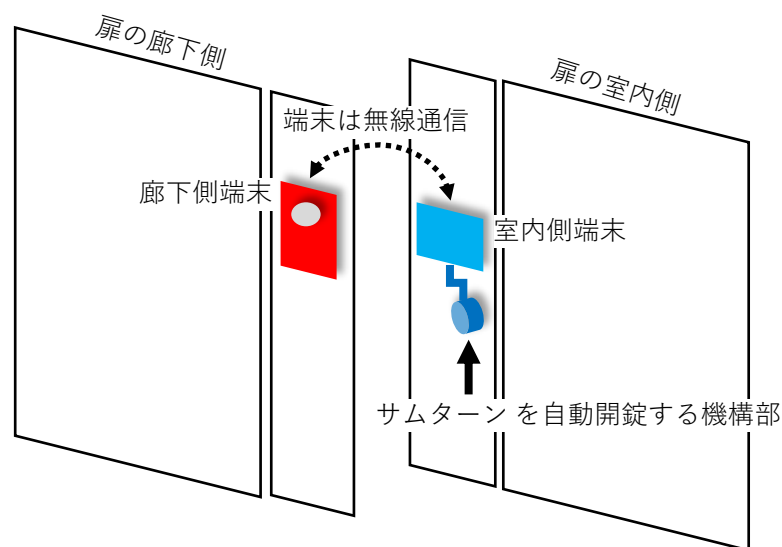


図 2.1 開発するシステムの全体像

開発するシステムは、学生証での開錠システムにアドオンして動作するもので、廊下側からは顔認識によって、室内側からはタッチパネルの操作によって、ドアを開錠する。本研究で開発したシステムはネットワークに接続しないが、将来的にネットワークに接続することで、現在の在室状態を遠隔から確認したり、月間・年間の在室時間などの統計情報を確認できるシステムに発展させることができる。

2.2 廊下側端末の構成

廊下側端末の機能は入室者の顔画像を認識し、登録された人物であれば、室内側端末に開錠の指示を無線で送信することである。廊下側端末の主なコンポーネントの名称を表 2.1 に、外観を図 2.1 に示す。

表 2.1 廊下側端末の主なコンポーネント

本体	Raspberry Pi Zero W
顔認識カメラ	オムロン社のHVC-P2
人感センサ	焦電センサ HC-SR501
無線モジュール	MONOSTICK-B

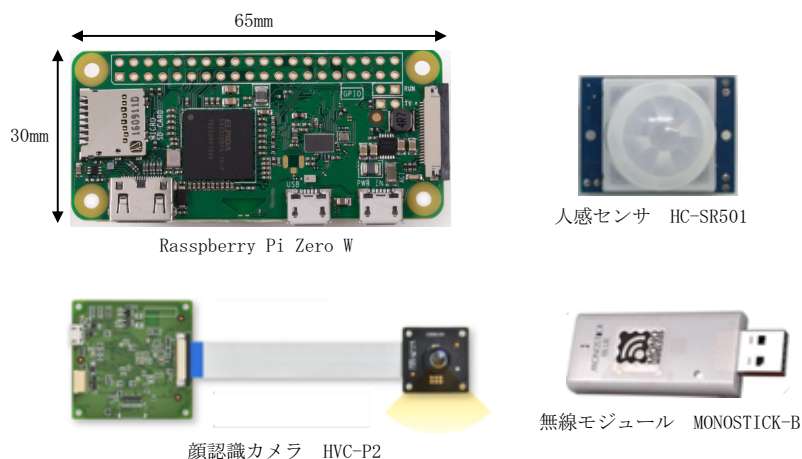


図 2.1 主要コンポーネントの外観

Zero W は最も小型の Raspberry Pi である Raspberry Pi Zero に、WiFi と Bluetooth の機能を付け加えたものである。

顔認識カメラ HVC-P2 は、オムロン社がヒューマンビジョンコンポという名称で開発・販売しているインテリジェントカメラモジュールである。HVC-P2 は、撮影した画像中の顔を検出することや、その顔の年齢・性別を推定すること、あらかじめ登録した顔画像との識別、などが可能である。Raspberry Pi と USB で接続し、HVC-P2 にコマンドを指令することで、これらの機能を利用する。

人感センサ HC-SR501 は焦電センサで、人体が発する熱を検出する。人体を検出した時に顔認証を実行することで、システムの電力消費を削減するために用いる。

無線モジュール MONOSTICK-B は、近距離無線通信規格である IEEE802.15.4 に準拠した無線モジュールで、低速、低コスト、低消費電力である。本システムでは、廊下側端

末が顔を認識したことを受けて室内側端末が開錠操作を行う時, この二者間で通信が必要であり, そのために利用した.

廊下側端末の各コンポーネントの接続状態を図 2.2 に示す. 人体感知センサを Zero W の GPIO(General Input Output, 汎用入出力)端子に接続した. Zero W には USB 端子が 1 つしかないため, 顔認識カメラと無線モジュールの接続には USB ハブを使った.

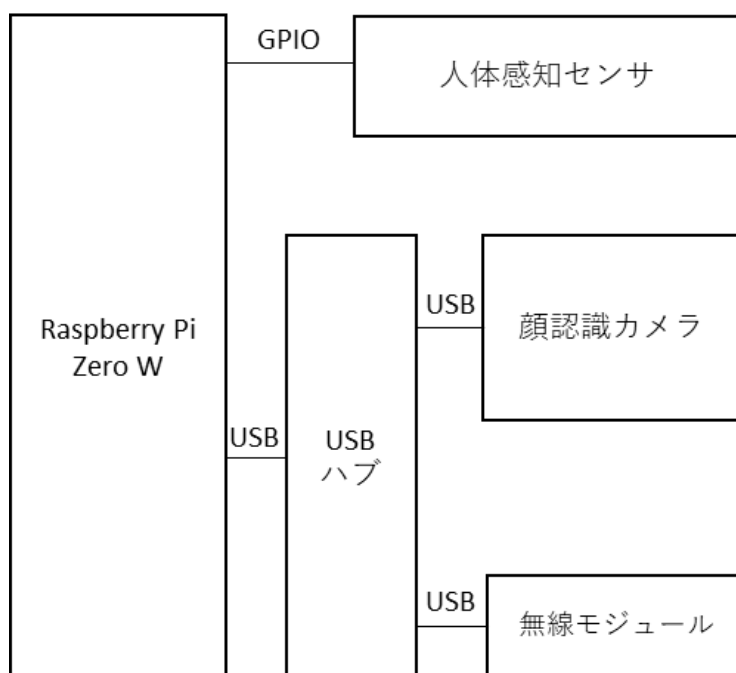


図 2.2 廊下側端末の接続を表したブロック図

2.3 室内側端末の構成

室内側端末は, 廊下側端末が送信した開錠の指示を受けて, 室内側にあるサムターンを回して開錠する. また, 開錠指令に含まれる個人 ID を取り出し, 入室ログを残す. この情報は室内側端末のディスプレイに表示し, 退出する際に, GUI に配置されているボタンをクリックすることによって開錠する. 廊下側端末の主なコンポーネントの名称を表 2.2 に, 外観を図 2.3 に示す.

表 2.2 室内側端末の主なコンポーネント

本体	Raspberry Pi 2 Model B
ディスプレイ	Raspberry Pi 7インチ 公式タッチディスプレイ
サーボモータ	GWS Servo micro 2BBMG
無線モジュール	MONOSTICK-B
サムターン開錠機構部	3Dプリンタで自作

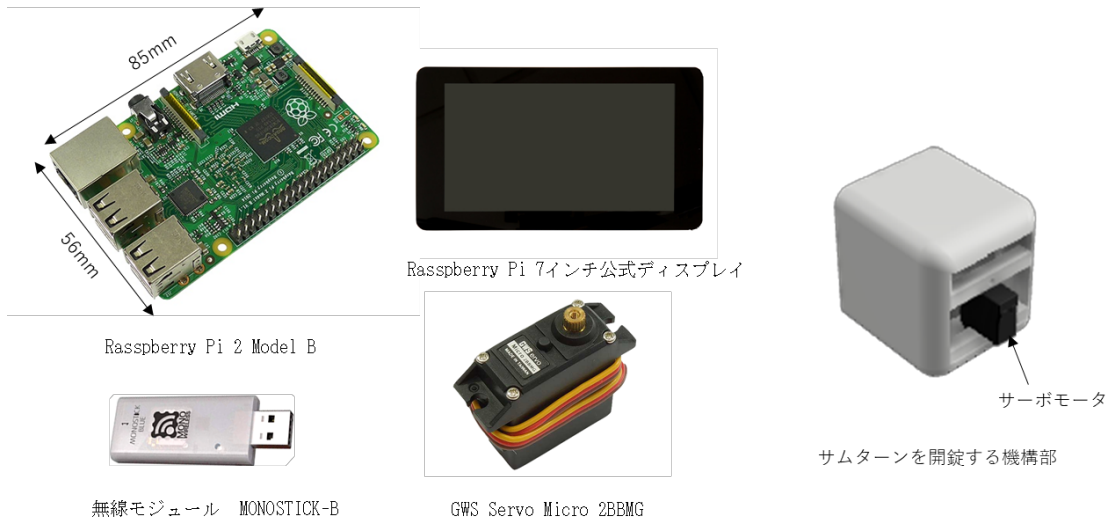


図 2.3 室内側端末の主要コンポーネントの外観

Pi2 は廊下側端末で動作している Zero W より少し大きな名刺サイズのワンボードコンピュータである。

Raspberry Pi 7 インチ公式ディスプレイはタッチパネルディスプレイである。GPIO かフレキシブルフラットケーブルで Raspberry Pi と接続する。本研究ではフレキシブルフラットケーブルで接続している。

無線モジュールは廊下側端末と同じ MONOSTICK をつかっており、廊下側端末から送信された情報を取得する。

室内側にあるサムターンを自動開錠する機構部は、サーボモータと 3D プリンタでプリントアウトした部品でできている。サムターンのノブを回すようになっている。

図 2.4 に室内側端末の主要部品の接続状態を示す。Pi2 は Zero W と同様に GPIO を備えている。開錠機構部のサーボモータは Pi2 の GPIO で制御する。Pi2 の電源ではサー

ボモータを動かすための電力が足りないため外部電源を用いた。無線モジュールは、廊下側端末と同じように USB で接続されている。

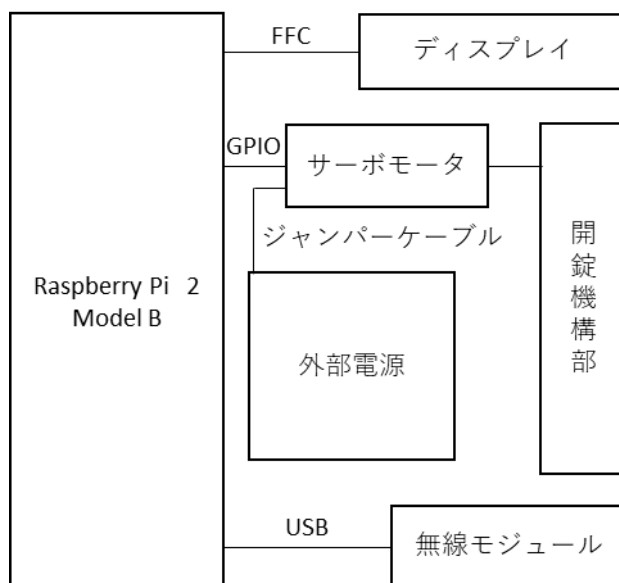


図 2.4 室内側端末の接続を表したブロック図

2.4 サムターンを開錠する機構部

この節では、サムターンを開錠する機構部の構造と動作原理を説明する。

2.4.1 構造

開錠機構部は 4 つのコンポーネントからできている。サーボモータ以外のコンポーネントは、メカニカル CAD である fusion360 を利用して作図し、3D プリンタ Afinia_H400 3D Printer を用いて作成した。開錠機構部の各コンポーネントの外観を図 2.5 に示す。

サーボモータ接合部は、図 2.5 の写真裏面にサーボモータの軸を取り付け、モータ軸の回転を伝える。サムターン接合部は、写真裏面にサムターンのノブを差し込む溝が加工されており、サムターンの動きと連動する。サーボモータ接合部とサムターン接合部の写真で見えている面を噛み合わせることで、サーボモータの動きをサムターンに伝える。これらの部品を外装部に装着し、外装部をサムターンが取り付けてあるドア面に固定する。

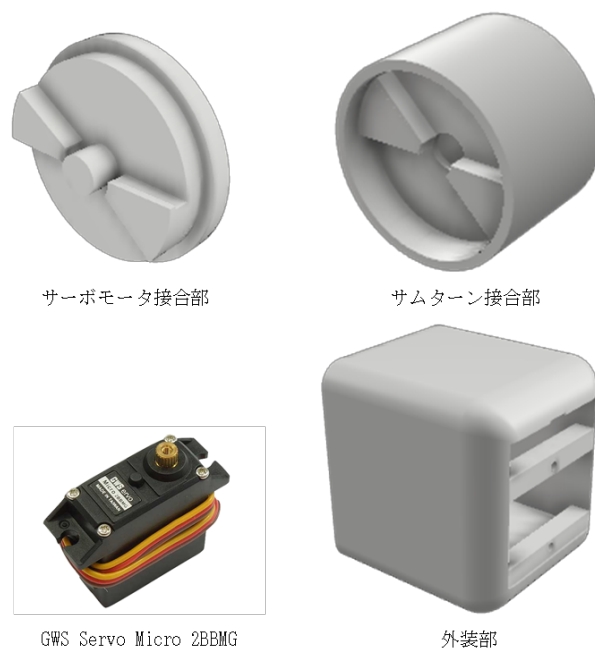


図 2.5 サムターンを開錠する機構部のコンポーネント

図 2.6 に、サーボモータ接合部とサムターン接合部の表裏の断面形状を示す。サーボモータ接合部の裏面に、サーボモータホーン（サーボモータの軸を接合面に取り付ける部品）をねじ止めする小さな穴を開けた。サムターン接合部の裏面に、サムターンを差し込む溝をつけた。両部品を組み合わせるために、図のようにくさび状の突起を加工した。サムターンを回転させて開錠する場合、サーボモータ接合部の突起がサムターン接合部の突起を押して回転させる。その後、数秒の間において、サムターンは自動的に元の位置に戻るのだが、それに先んじてサーボモータを元の位置に戻す。このようにすることで、サムターンが復帰する場合に、サムターン側に負荷をかけないように工夫した。

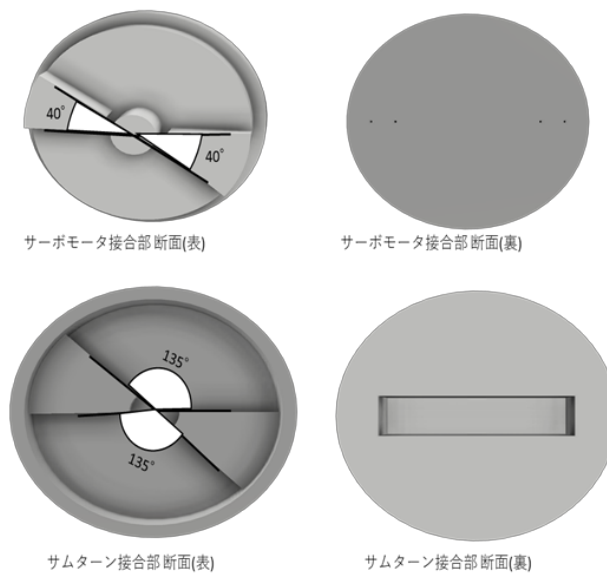


図 2.6 駆動部の各部品の断面

図 2.7 に、これらの部品の組み合わせ方を示す。

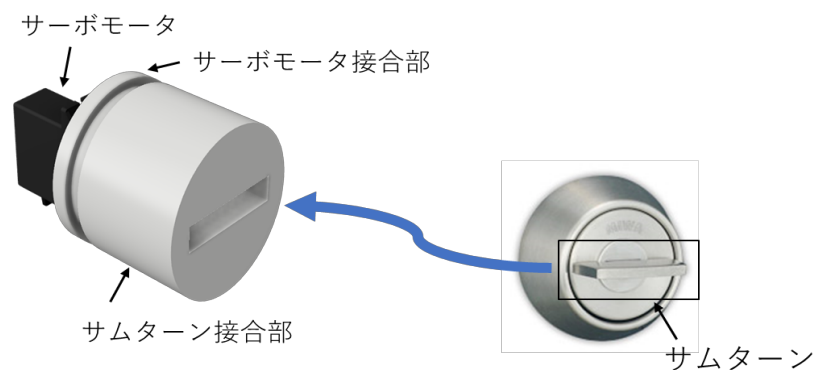


図 2.7 駆動部の接続状態

2.4.2 動作原理

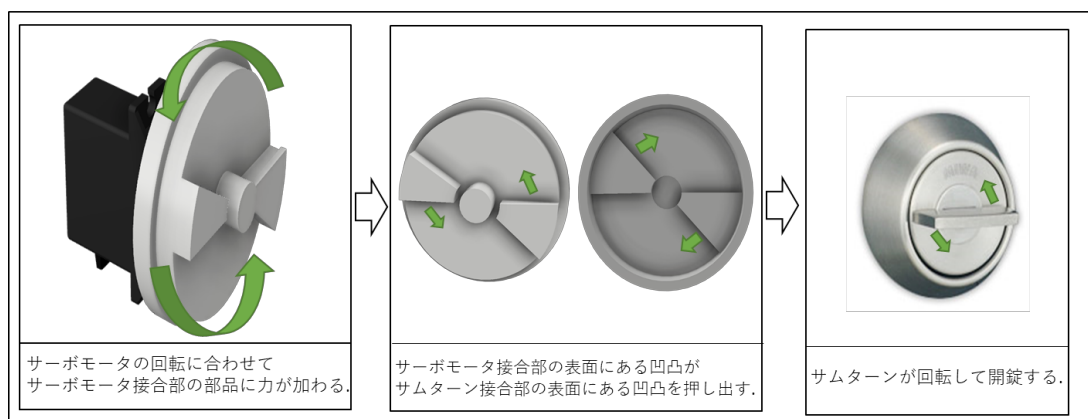
本システムを使った開錠・施錠の動作シーケンスは次のようになる。

- ① サムターンを開錠側に回転させて開錠する。
- ② 人がドアを開けて入退室する。
- ③ ②の動作が行われている間に、サーボモータを元の位置に復帰させる。
- ④ サムターンが施錠側に回転して施錠される（これは、元のシステムの動作）。

ここで、サムターンを開錠側に回転させる動作は開発するシステムのサーボモータで行う。施錠側に回転させる動作は、元のシステムが自動的に行う。

図 2.8 の上側は開錠時の動作を示す。サーボモータ接合部のくさび状突起がサムターン接合部のくさび状突起を押すことで、サムターンを開錠方向に回転させる。図 2.8 の下側が施錠時の動作を示す。サムターンの施錠方向への回転は、元のシステムが行う動作であるから、これを妨げないように、先んじてサーボモータの位置をもとに戻している。

サムターンを開錠するまでの動作



開錠後の動作

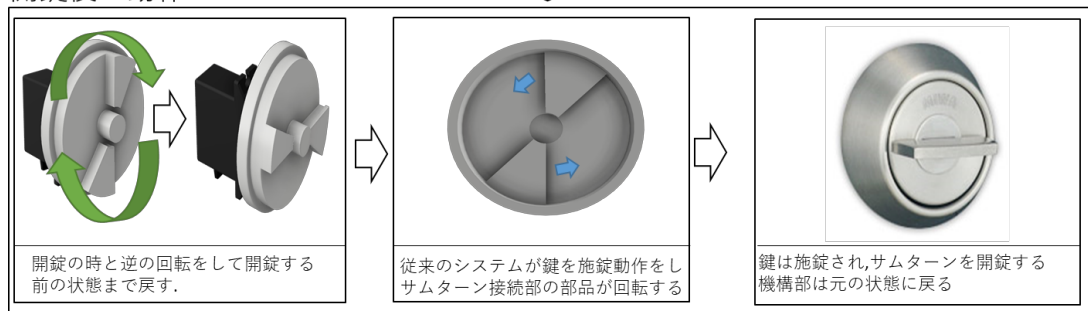


図 2.8 駆動部の動作原理

3 章 自動入退室システムのソフトウェア

本章では、廊下側端末と室内側端末の動作を、それぞれのフローチャートを示しながら説明し、最後に全体としての動作を説明する。

3.1 廊下側端末

まず、廊下側端末の主なコンポーネントである顔認識カメラについて説明し、その後廊下側端末全体の説明を行う。

顔認識カメラは廊下側端末に接続される。顔認証を行う場合、事前にアルバムデータとよばれる顔データ作成し、そのデータを顔認識カメラのフラッシュ ROM に保存しておく必要がある。アルバムデータは、登録した全員の顔のデータが集まっているデータである。アルバムデータを作成するには、個人を区別する ID（本研究ではこの ID のことを Face ID と呼ぶ）が必要である。本研究では、指導教員である蚊野教授を 0 番とし、学年に分けて名前順に Face ID を割り当て、アルバムデータを作成した。研究室メンバーに登録されている Face ID を表 3.1 に、アルバムデータを登録する流れを図 3.1 のフローチャートに示す。

表 3.1 研究室メンバーと ID の関係

名前	ID
蚊野 教授	0
田中	1
富永	2
西島	3
平山	4
青木	5
斉木	6
白石	7
西岡	8
花木	9

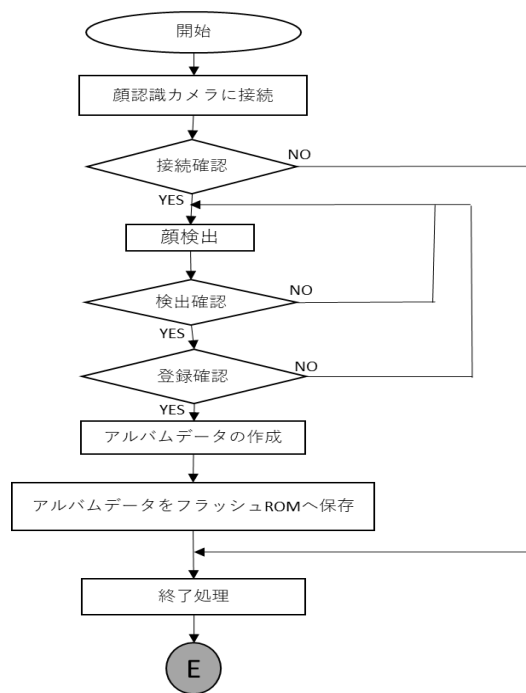


図 3.1 顔画像を登録するまでの動作

顔認識カメラにアルバムデータが登録できると、顔認証が可能になる。顔認証を行うフローチャートを図 3.2 に示す。顔認証の結果、Face ID とその信頼度を顔認識カメラから受け取ることが可能になる。

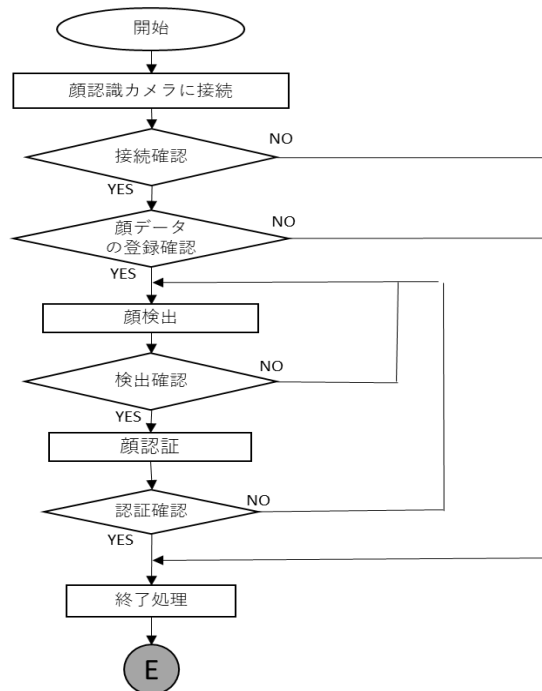


図 3.2 顔認証の動作

廊下側端末全体としての動作は、① 人体感知センサを用いて人体を検出、② 顔認識カメラを用いて顔認証を実行、③ Face ID と信頼度で開錠の判断を行う、④ 無線モジュールを使い室内側端末に対し解錠指示を送信する、という流れである。その動作のフローチャートを図 3.3 に示す。

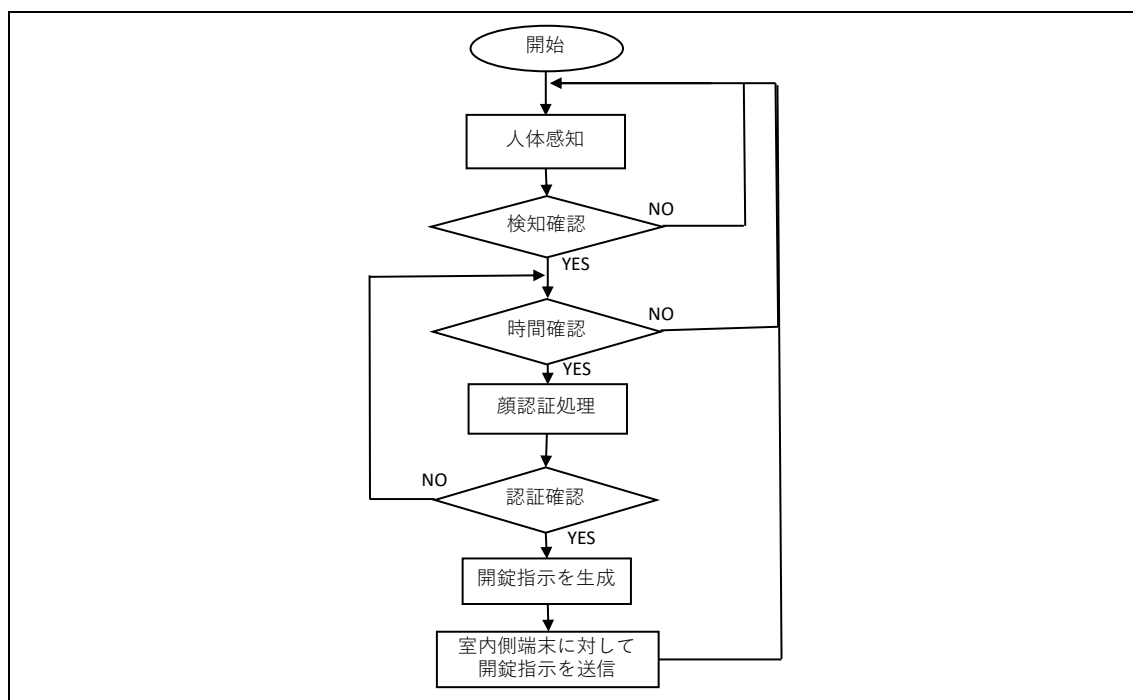


図 3.3 廊下側端末の動作

ここで注意点が 1 つある。廊下側端末に接続している顔認識カメラと無線モジュールは USB で接続されている。この 2 つの接続は Raspberry Pi 内部では、両方がシリアル通信を用いて動作をするという点である。それぞれがシリアル通信を行っている時、片方の通信の途中で他方が通信を行うと予期せぬエラーが発生する。完全に通信を終わらせる必要があるため、プログラムでは注意が必要である。

3.2 室内側端末

室内側端末の動作を説明する前に、ディスプレイに表示する GUI について説明する。作成した GUI は、ホームウィンドウと、メンバーごとに分かれた複数のユーザウィンドウの 2 種類からできている。

ホームウィンドウで学年を選択するとユーザウィンドウに遷移する。ユーザウィンドウは学年ごとにユーザのパネルが配置されている。退室する際には、個人の写真画像の下に配置された退室ボタンをクリックすることで退室するようになっている。退室ボ

タンは、在室状態をまとめたデータ(以下このデータのことを Now データとよぶ)を参照して表示するため、在室していない学生は退室ボタンを押せない。図 3.4, 図 3.5 に GUI の構成図, 図 3.6 にウィンドウを遷移するためのフローチャートを示す。

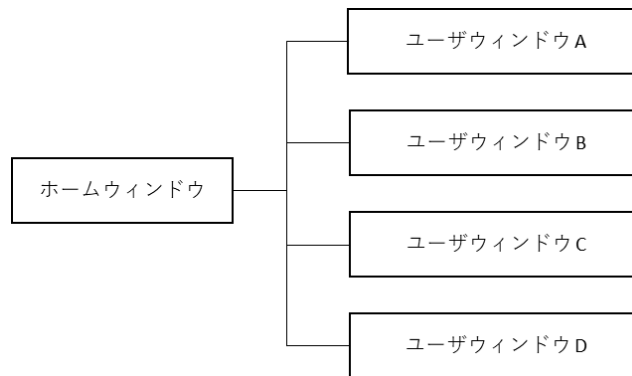


図 3.4 GUI の構成図

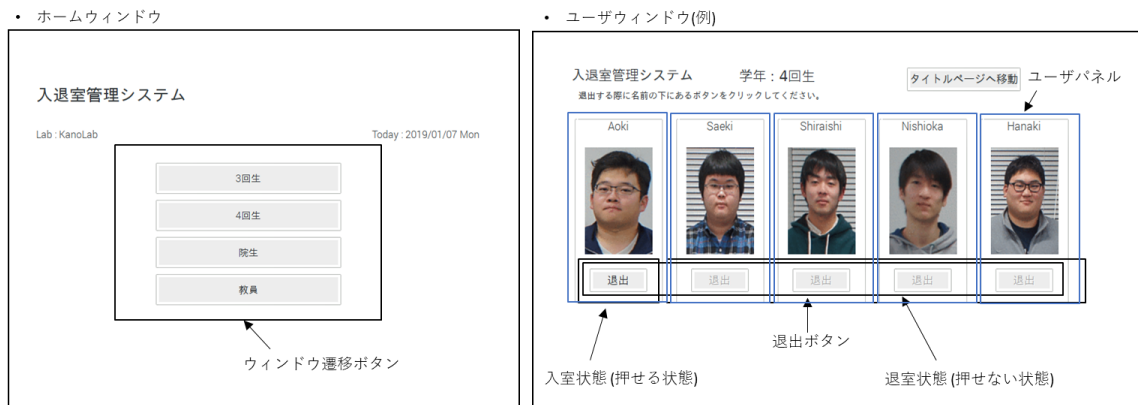


図 3.5 GUI の外観

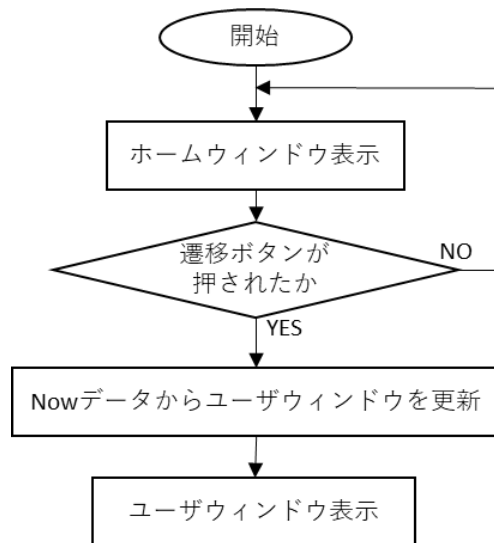


図 3.6 GUI でユーザウィンドウを表示するまでの流れ

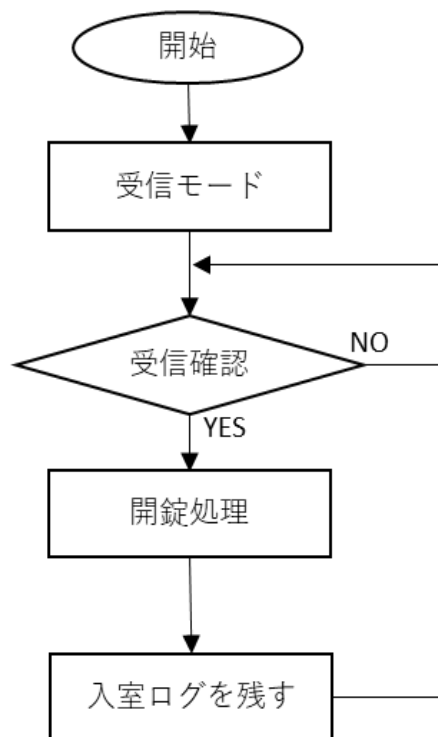
退室ボタンを押すと，図 3.7 のように確認ウィンドウが開く．Yes ボタンを選択することでサムターンが回り，開錠され退室できる．



図 3.7 退室ボタンを押した場合

次に，室内側端末の動作を説明する．室内側端末は，廊下側端末から開錠指示が送られてきた時と，室内の人が退出する時に動作する．これら 2 つの動作のフローチャートを図 3.8 に記す．2 つの動作は同時に発生する可能性があるため，マルチスレッド処理を行うようにプログラミングした．ただし，いずれにおいても，サムターンを開錠する動作が発生する可能性があり，サムターン開錠の要求が同時に発生した場合，排他的な動作として動くようにプログラミングした．

入室時の動作



退室時の動作

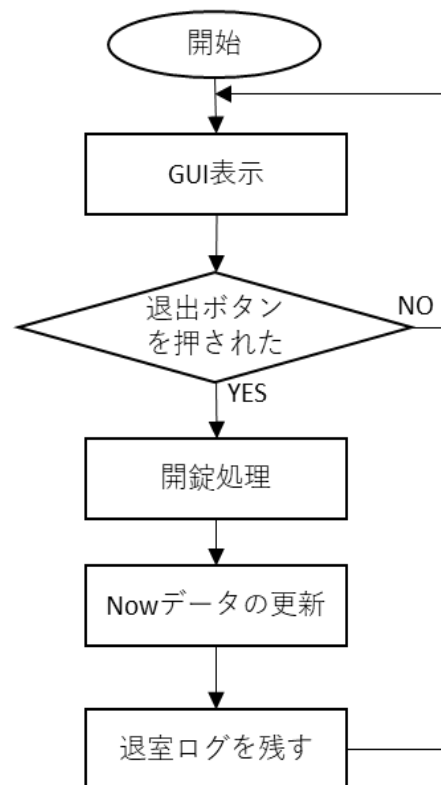
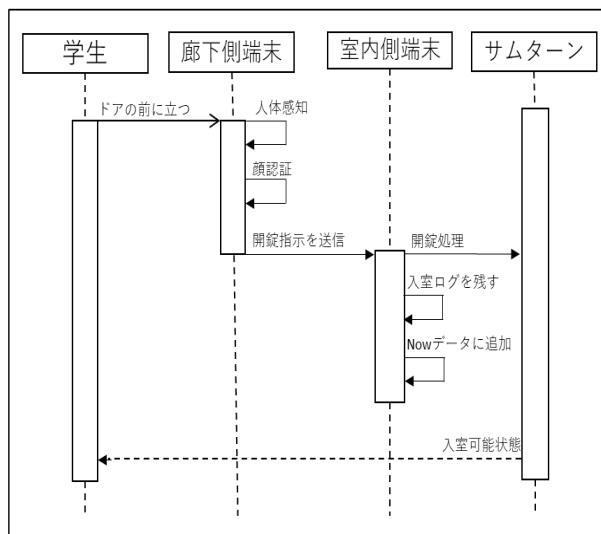


図 3.8 室内側端末のフローチャート

3.3 システム全体の動作

構築したシステムが時間経過とともに，どのように動作をするかをまとめたシーケンス図を図 3.9 に示す.

• 入室時



• 退室時

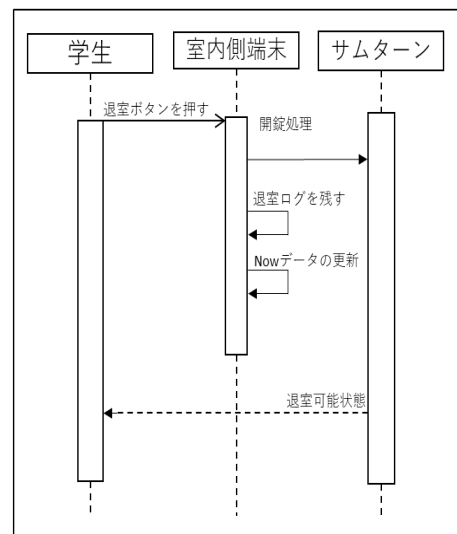


図 3.9 システム全体の動作の流れを示すシーケンス図

このシステムを動作させることにより，メンバー名，在室状態，日付，時間の4つの情報を持つログが生成される．システムが生成する入退室ログの例を図 3.10 に示す．

```

State : in Name: Hanaki Date: 2018/11/27 Time: 13:20:40
State : in Name: Hirayama Date: 2018/11/27 Time: 14:30:20
State : in Name: Kano Date: 2018/11/27 Time: 15:02:37
State : out Name: Kano Date: 2018/11/27 Time: 15:29:57
State : out Name: Hirayama Date: 2018/11/27 Time: 15:30:17
State : in Name: Kano Date: 2018/11/27 Time: 15:34:33
State : out Name: Kano Date: 2018/11/27 Time: 15:50:57
  
```

図 3.10 本研究で開発したシステムの入退室ログの例

4 章 自動入退室システムの動作検証と考察

開発した自動入退室システムの動作を検証した．この章では性能, 信頼度などについて記述する．

4.1 性能検証

開発した自動入退室システムの動作を検証するために, 入室動作, 退室動作など, 個別的な動作について確認した．

4.1.1 入室動作

まず, 顔認識カメラ単独での顔認証の動作について確認した．学生実験室内で研究室のメンバー5人の顔画像を登録し, 同じ場所で個人認証を行なった．顔認識カメラの出力データを確認した結果を表 4.1 に示す．正しい Face ID が出力されていることと, その信頼度として 810 ～ 1000 の値が得られていることがわかった．信頼度の最大値は 1000 である．この結果から, 顔認識カメラを使った個人認証が正しく実行できていることが確認できた．

表 4.1 顔認証の結果（顔の登録と認証を学生実験室内で行なった場合）

氏名	正しいID	結果 ID	信頼度
蚊野教授	0	0	910 ～ 1000
斎木	6	6	917 ～ 981
白石	7	7	810 ～ 978
西岡	8	8	893 ～ 985
花木	9	9	902 ～ 988

次に, 開発したシステムをドアに取り付け, 入室動作を検証した．その結果, 入室できるメンバと入室できないメンバが存在した．入室動作を確認した様子を図 4.1 に, 検証した結果を表 4.2 に示す．

- 室内側端末



- 室外側端末



図 4.1 動作検証の様子

表 4.2 顔認証の結果（顔の登録を設置場所で行なった場合）

氏名	結果
蚊野教授	入室不可
斎木	入室不可
白石	入室不可
西岡	入室可能
花木	入室可能

認証に失敗する原因を調査すると、設置場所が暗く、信頼度の数値が 100 ～ 200 程度下がるのが原因であった。そこで、実際に動作する環境に合わせて顔画像の再登録を行った。その後、認証時の信頼度の数値を確認すると、表 4.3 のように十分な数値になっていた。また、検証作業中に収録したログデータの例を図 4.2 に示す。

表 4.3 入室動作の再検証の結果

氏名	正しいID	結果 ID	信頼度	結果
蚊野教授	0	0	970 ～ 998	入室可能
斎木	6	6	987 ～ 1000	入室可能
白石	7	7	968 ～ 1000	入室可能
西岡	8	8	976 ～ 1000	入室可能
花木	9	9	978 ～ 1000	入室可能

```

State : in Name: Nishioka Date: 2019/02/05 Time: 16:10:12
State : in Name: Saeki Date: 2019/02/05 Time: 16:50:17
State : in Name: Shiraishi Date: 2019/02/05 Time: 16:50:33
State : in Name: Hanaki Date: 2019/02/05 Time: 16:48:49
State : in Name: Kano Date: 2019/02/05 Time: 17:01:22

```

図 4.2 入室動作の検証作業中に収録したログデータ

4.1.2 退室動作

退室動作について, サムターンを正しく回転して開錠できたことと, 正しい退室ログが残っていること確認した. 退室動作の検証結果を表 4.4 に, 検証作業中に収録したログデータを図 4.3 に示す. これらから, 退出動作が正しく作動したことが検証できた.

表 4.4 退室動作の検証結果

氏名	結果
蚊野教授	退室可能
斎木	退室可能
白石	退室可能
西岡	退室可能
花木	退室可能

```

State : out Name: Saeki Date: 2019/02/05 Time: 16:58:45
State : out Name: Nishioka Date: 2019/02/05 Time: 17:00:07
State : out Name: Kano Date: 2019/02/05 Time: 17:05:32
State : out Name: Shiraishi Date: 2019/02/05 Time: 18:20:25
State : out Name: Hanaki Date: 2019/02/05 Time: 18:22:50

```

図 4.3 退室動作の検証作業中に収録したログデータ

4.1.3 総合的な評価

開発したシステムを既存のカードキーによる入退室システムにアドオンして動作させた. 入退室の基本動作は機能したが, 完全に製品レベルの動作というわけではなく, 良い点・不足な点が見られた. それらを以下に示す.

- ① 顔をかざすだけで入室できることは, カードキーを使った入室よりも便利である.
- ② ディスプレイの GUI を操作することによる退出は, サムターンを回すよりも不便で

ある。ただしこれは、ディスプレイのタッチパネルが動作しなかったためマウスで GUI を操作するようにしたことが関係している。タッチパネル操作であれば印象が変わると思われる。

- ③ 顔画像を使った個人認証は、なりすましが比較的容易である。一方で、照明に変動や顔画像の変化に弱いので、セキュリティが弱いと言える。

4.2 考察

検証の過程で、開発したシステムに3つの問題があることがわかった。以下にその問題点と解決の方法を示す。

- ① 顔画像を登録した場所とシステムを設置する場所の照明環境が異なると、顔認証の信頼度が下がるという問題があった。この問題を解決するは、顔認証を行う場所の明るさを、照度センサ等を用いて計測し、認証の閾値を調整する方法や、顔認証の際に利用者の顔を照明で明るくし、顔画像の入力を安定化する方法が考えられる。
- ② 廊下側端末は、スリープ状態、人体を検出した状態、顔画像の認証を行なっている状態、認証に成功あるは失敗した状態、という複数の状態を遷移する。しかし、ユーザが現在の状態を確認することができないので、不安になるという問題があった。この問題を解決するには、LED 等を用いて現在の動作状態を確認できるようにすれば良い。
- ③ 廊下側端末は 10.7Wh の USB モバイル電源を使用しており、連続動作させると半日ほどで電池を使い切ってしまうという問題があった。この問題を解決する一つの方法は AC 電源を使うことである。しかしこの方法では AC ラインが供給されている場所でしか使うことができない。他の解決策はシステムを省電力化することである。現在と同じコンポーネントを使うシステムであっても、可能な限り省電力で動作するようにプログラムすることで電池寿命を伸ばすことができると考えている。

5 章 結論

研究目的で開発したシステムであるが, 入退室の実利用場面で正しく動作させることができた. カードキーを持ち歩く必要が無くなり, 入室時の利便性は向上した. 一方, まだ多くの課題が残っていることも分かった. 課題の中には, 顔認証はセキュリティ面では弱いということや, ログを室内側端末でしか保存していないため, 後からログを確認する場合には室内側端末から確認する必要があるといったものもある.

以上, 本研究で開発したシステムは, 動作はするが, まだ実用的に動作させるには乗り越えなくてはならない課題が多く残っているという状態である. 運用するシステムの基盤を作ったという結果であった.

参考文献

- [1] 大澤 文孝 「TWE-LITE で始める簡単電子工作」，株式会社工学社 平成 26 年 8 月
20 日 初版発行
- [2] 大澤 文孝 「TWE-LITE で始める「センサー」」，株式会社工学社 平成 27 年 11 月
25 日 初版発行

謝辞

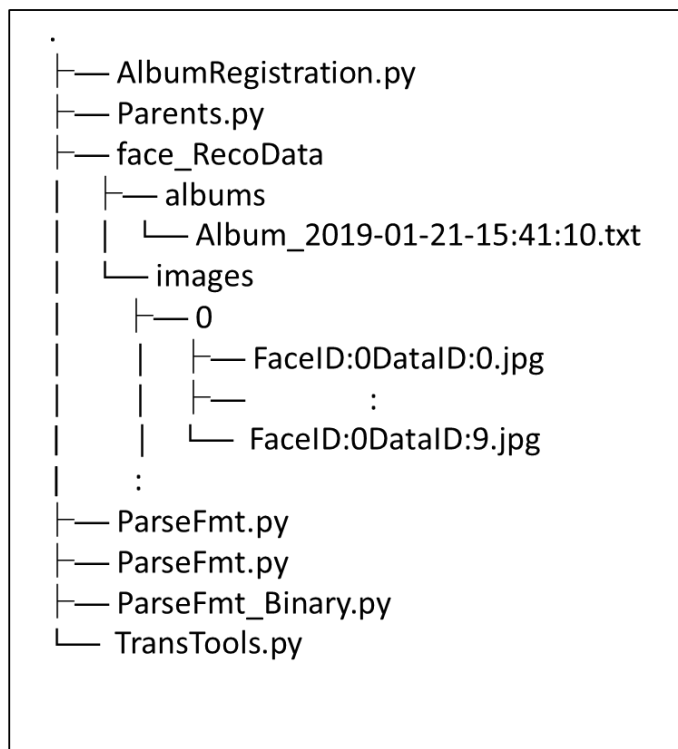
本論文の作成に当たり,丁寧なご指導をしてくださった蚊野浩教授に深く感謝申し上げます.

また,本研究を行うに当たり被験者となっていただきました蚊野研究室のメンバー,そして激励をいただきました友人に対し感謝の意を表します.

付録 1 本研究で開発したファイルの構成

本研究で開発したファイルをデバイス毎に分けると以下の図のようになる。

- 廊下側端末



- 室内側端末

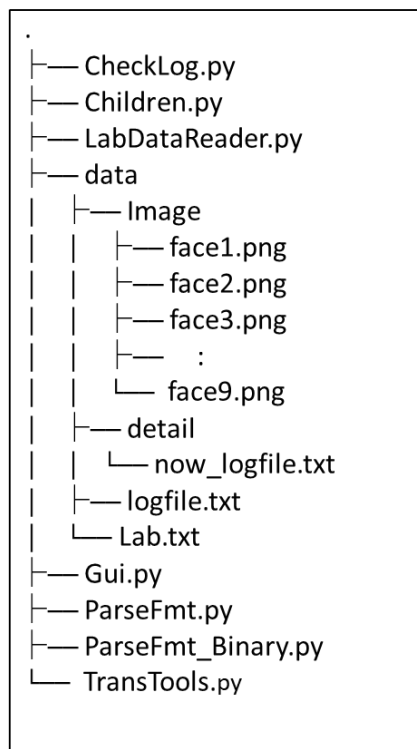


図 付録-1 本研究で開発したファイル

プログラムの開発において、顔認識カメラを制御するためにシリアル通信で送信しているコマンドの仕様書

(https://omronfs.omron.com/ja_JP/ecb/products/pdf/B5T-007001_CommandSpecifications_JP_B.pdf)を参考にした。無線モジュールの制御をしているプログラムは, MONO WIRELESS 社のページに載っている Raspberry Pi カメラ画像転送プログラム

(https://mono-wireless.com/jp/products/TWE-APPS/App_Uart/raspi_camera/index.html)を参考にした。各デバイスの MONOSTICK のファームウェアを、標準で入っている「超簡単！標準アプリ」から、カメラ画像転送プログラム用のファームウェアを利用し、廊下側端末には Parents.bin, 室内側端末には Child.bin, に変更した。

- 廊下側端末

[プログラム名]

Parents.py

[内容]

廊下側端末のメインプログラム.

[プログラム名]

ParseFmt_Binary.py

[内容]

無線モジュールを用いて送られてきた信号のバイナリ書式の解読を行うプログラム

[プログラム名]

ParseFmt.py

[内容]

書式パーサーである parseFmt_Binary.py の基底クラス

[プログラム名]

TransTools.py

[内容]

シリアルポートの開閉や, 無線モジュールを送信した後に ACK(肯定応答)の確認, 無線モジュールで送信を行う開錠指示の生成等を行なっている.

[プログラム名]

AlbumRegistration.py

[内容]

顔認識カメラを用いてアルバムデータ生成登録を行うプログラム. はじめに顔検出を行うようになっており, 検出状態は OpenCV の関数を使い表示している. 登録したい顔が検出されている状態で” q キー” 押すことで顔検出が終了し, その後, ID と何枚目の画像か入力することで検出した顔画像で指定の ID でのアルバムデータの生成を行い, 顔認識カメラのフラッシュ ROM へ登録を行うようになっている. 登録した顔画像とアルバムデータは作業ディレクトリの中にある face_RecoData 内の, アルバムデータは albums フォルダに, そのアルバムデータに保存されている顔画像は images/(ID の番号) フォルダの場所に保存されるようになっている.

- 室内側端末

[プログラム名]

Children. py

[内容]

室内側端末でのメインプログラム.

[プログラム名]

ParseFmt_Binary. py

ParseFmt. py

TransTools. py

[内容]

無線モジュールを操作するために用意したプログラム. (廊下側端末に入っているものと同じもの)

[プログラム名]

CheckLog. py

[内容]

GUI を表示する際の元となっている Now データ (data/detail/now_logfile. txt) の読み込みや, 退出があった場合は変更等を行っている。

[プログラム名]

LabDataReader. py

[内容]

GUI を表示する際に用意した研究室の情報がまとまっているデータ (data/Lab. txt) の読み込み等を行い, GUI を表示するために必要なデータの用意を行っている。

[プログラム名]

Gui. py

[内容]

GUI の表示を行っている. GUI を表示するにあたって, クロスプラットフォームなウィジェット・ツールキットである wxPython, 本研究で作成した LabDataReader. py, CheckLog. py, あらかじめとっておいた研究室の各メンバーの顔画像データ (data/Image/face (ID). png) を用いている。

付録2 システムブート時に指定のプログラムを実行する方法

廊下側端末はディスプレイを使わずに動作させるため、システムがブートしたときに、自動的にアプリケーションプログラムを実行している。その動作をさせるために、システム管理のソフトウェアである `systemd` を利用した。

`Systemd` を利用してブート時に指定のプログラムを実行するには、プロセスの起動/停止に関する設定を記述した Service ファイルを準備することと、Service ファイルを自動実行させる操作、の2つを行う必要がある。

● Service ファイル

Service ファイルは `/usr/lib/system/systemd` か `/etc/systemd/system` に保存する。

(注) 同じ名前の Service ファイルが両方のフォルダにある場合、`/etc/...` のファイルが優先される。

Service ファイルの例を以下に示す。

```
[Unit]
Description=Execute the python file sample.py that exists in the home directory
[Service]
ExecStart=/usr/bin/python /home/pi/sample.py
[Install]
WantedBy=multi-user.target
```

上記の Service ファイルは、ホームディレクトリに存在する `sample.py` を自動的に実行する場合に作成する Service ファイルである。設定を行うファイルは `Unit`、`Install`、`Unit` タイプ固有のセッション (例の Service ファイルの場合は `Service`) の3つのセクションに分かれている。`Unit` セクションは `Unit` の依存関係/順序関係を設定する項目、`Install` セクションは `systemctl enable / disable` コマンドの動作を設定する項目、タイプ固有のセクションはそのタイプ固有の設定を行う項目である。

例として記述した Service ファイルの説明を行う。`Unit` セッションの `Description` オプションは `Unit` の説明を行っている。`Service` セッションの `ExecStart` オプションはサービス起動のコマンドを記述しており、`/usr/bin` の中にある `python` コマンドを用いて、ホームディレクトリの `sample.py` を実行するように設定を行っている。`Install` セッションの `WantedBy` オプションは `enable` 時にこの `Unit` の `wants` ディレクトリにリン

クを作成するオプションであり, 一般的なサービスの場合は例に記述したように multi-user.target を記述しておけばよい. 各セッションには説明したオプション以外のものも存在するので調べてみると良い.

● Service ファイルを自動実行させる方法

作成した Service ファイルを自動実行させるには, 以下のコマンドを使用してシンボリックリンクを張る.

```
$ sudo systemctl enable sample.service(作成したサービスファイル)
```

(Tips)

自動実行を止めるには以下のコマンドを入力する.

```
$ sudo systemctl disable sample.service(作成したサービスファイル)
```

どのように動作するかを確認したい場合は以下のコマンドを入力する.

```
$ sudo systemctl start sample.service(作成したサービスファイル)
```

付録 3 使用上の注意点

本研究ではサーボモータを使った。サーボモータはPWM(Pulse Width Modulation)で制御され, そのパルス幅で回る角度の設定を行う。PWM 信号の例を図 付録-2 に示す。

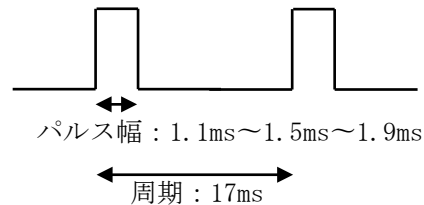


図 付録-2 PWM 信号

ソフトウェアで PWM 信号を発生させた場合, OS の動作などが作用し, 周期に誤差が生じる。そこでハードウェア PWM を使った。python からハードウェア PWM を呼び出すことができるライブラリである pigpio(<http://abyz.me.uk/rpi/pigpio/>)を使った。pigpio ライブラリは, ラズベリーパイの GPIO にアクセスするために, 通常, 次のように実行する。

```
$ python aaa.py(pigpio ライブラリを用いたプログラム)
```

しかし, 上記のコマンドではエラーが発生する。pigpio ライブラリを用いたプログラムを実行する場合は

```
$ sudo pigpiod
```

のコマンドを実行し, そのあとで通常の実行コマンドを打つ必要がある。