

10 NFCを使ったスマートロックによる入退室管理

小原 朋也

指導教員 佐々木 建

1はじめに

1.1 研究の背景と目的

現在,モノがインターネットに接続されるようになり,身近でIoT(Internet of Things)の活用シーンを見かけることが多くなってきた.その様な中で,私も毎日の生活の中で何かIoTで便利にできることないかと考えたところ,学校の実習室等の鍵をカード式のスマートロック型にすれば入退室を楽にできるのではないかと考え,このテーマを選定した.

1.2 研究の意義

NFC(Near Field Communicationの略:近距離無線通信カードリーダ)によって検知したカードから情報を取得し,自動でデータを更新することや,その情報によりサーボモータを制御し,自動でサムターン式の鍵の開閉動作をさせること等,組み込み系の醍醐味をこの研究でさらに実感したこと,また,授業で習ったサーバやデータベース(DB)に関しての知識を,実際に使用して確認することで,さらに実用性についても理解を深めることができが研究の目的である.

2 研究概要

2.1 機能概要

NFCに,FeliCa(非接触型ICカード)をかざすとサーボモータを回して,鍵を開閉することを基本とする.カードの判別方法については,DBに登録されているIDm(固有のID番号)であれば入室を許可し,登録されていない場合はエラー表示をし,入室ができないことを示す.

DB登録された者で最初に操作した場合は,カードをかざすと開錠され,入室が可能となる.次に入室する者もカードをかざすことにより,DBに入室

者情報が追加されていく.誰が入退室したかは,管理者がWeb上から確認することができる.

LCDディスプレイには,入室している人数を表示させている.

退室する際も入室時同様にNFCにカードをかざして退室する.退室者の情報はDBから自動的に削除され,表示される在室中の人数も減るようにしている.在室の最後の者がカードをかざして在室者ゼロになった場合に施錠を行う.また,ドア内側のサムターン部分が装置で覆われてしまうため,緊急的に内側からの開錠する機能も実装した.

2.2 仕様詳細

今回のシステムは主にボタン読み取り・カードリーダ読み取り・Webページの表示機能で構成した.詳細は次に示す.

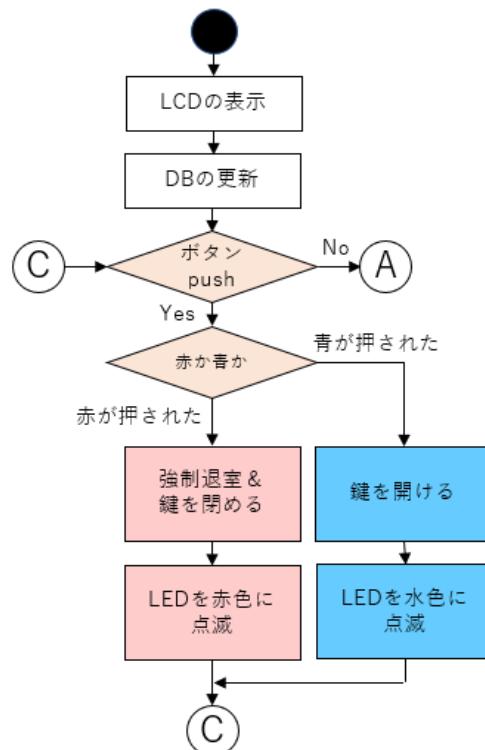


図1 ボタン部分のアクティビティ図

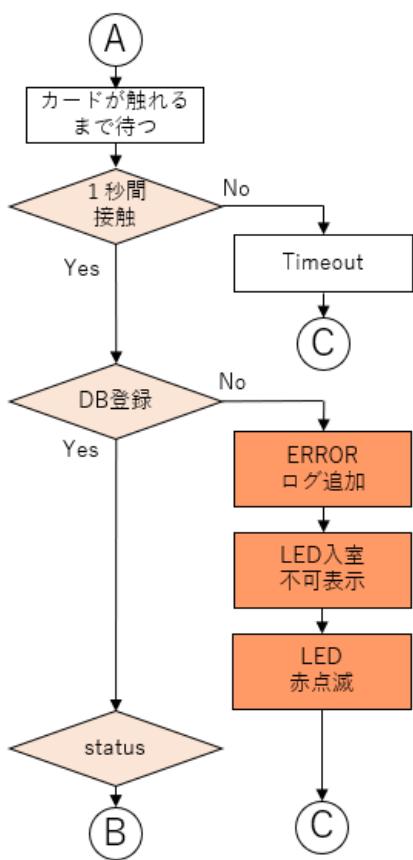


図2 カードリーダアクティビティ図(1)

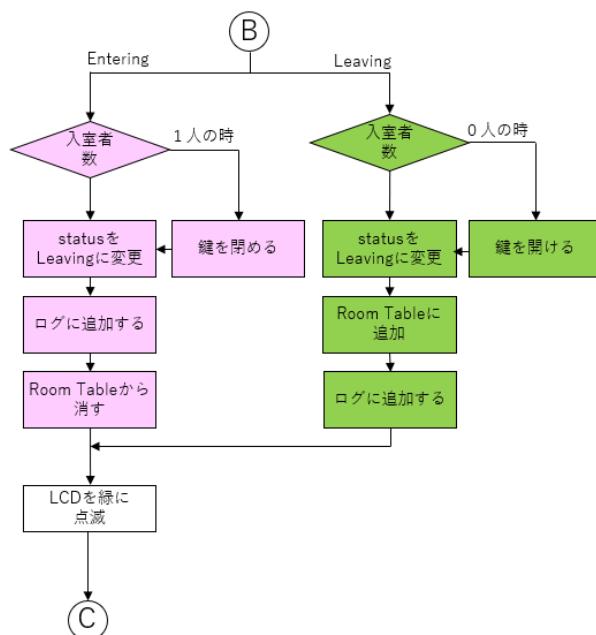


図3 カードリーダアクティビティ図(2)

2.2.1 ボタンの処理について

今回制作したものはサムターン部分にマウント部品を取り付けたため、手動での操作が困難なた

め、開錠や施錠のためのボタンを実装した。

ボタンの役割は次のとおりである。

- ・青ボタン 内側から鍵を開錠
- ・赤ボタン 内側から鍵を施錠

鍵の開閉については、在室の人数により動作が異なる。鍵を施錠するためには、自分が最後の1人になる必要がある(DB上では自分以外のデータがないことを指す)が、入室した者が退室時にカードをかざし忘れた場合、自分以外のデータが残っているため施錠したくても、施錠できないケースが発生する。その場合に使用する赤ボタンの役割は、在室データをすべて退室状態(クリア)にして、10秒後に施錠する機能を実装した。

2.2.2 カードリーダの処理について

カードリーダにカードをかざすことによって鍵の開閉を行うのが基本動作である。

在室を示すために使用したDBの各テーブルの構造(図5)を示す。

Person		Room		Log	
PK	p_id	PK	idm	PK	id
	idm		name		date
	name		entry_time		name
	status				time

図5 各テーブルの構造

[Person]テーブル

登録するカードのID、登録者の名前、入退室の状態のアтриビュートで構成される。

[Room]テーブル

入室者のみのレコードを管理し、カードのID、名前、入室時間のアтриビュートで構成される。

[Log]テーブル

入退室の記録をとったものであり、入退室日、名前、入退室時間のアтриビュートで構成される。

具体的なカードでの処理は次に示す。

(1)カードが触れたか

最初にカードがリーダに触れたかどうかの判定を行い、触れていない場合は、タイムアウトを1秒

設定しているため,DB の値読み取りから再度実行しなければならない。

(2) カードがDBに登録されているか

カードリーダで情報を読み取った場合,読み取ったIDがDBに登録されているかを判定する。登録者のデータは[Person]テーブルに保存されており,登録されていなかった場合はエラー処理を行う。

(3) status欄の判定

[Person]テーブルに登録されているカードの場合, status欄を判定する。status欄が「Leaving」の場合は入室の処理。status欄が「Entering」の場合は退室の処理に遷移する。

status欄の「Leaving」と「Entering」は現在, 部屋の外にいるか, 中にいるかを表している。カードリーダにかざすことで,「Leaving」と「Entering」が入れ替わるようになっている(図6)。

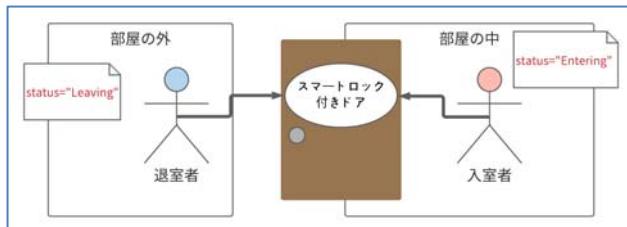


図6 利用者のユースケース図

(4) 入室の処理

入室する場合,[Room]テーブルのレコードが0件の場合(最初)のみ開錠する処理をしている。二人目以降はカードをかざす都度に,在室情報が更新([Person]テーブルのstatus項目を「Entering」に変更,[Room]テーブルおよび[Log]テーブルにレコードの挿入を行う)される。

(5) 退室の処理

退室する場合,[Room]テーブルのレコードが1件の場合のみ施錠する処理をしている。最後の一人以外は退室の際の施錠は行われない処理をしている。[Person]テーブルのstatus項目を「Leaving」に変更,[Log]テーブルにレコードを挿入,[Room]

テーブルからレコードが削除される。

2.2.3 LEDとLCDの表示パターンについて

ユーザが視覚的に今の状況を把握するためには,LCDディスプレイ(16文字×2行表示)と1つのLEDで複数の色を表現できるRGBフルカラーLEDを使用した。詳細については次に示す。

(1) LCDディスプレイ

2行表示の1行目には、「Please Touch.」を表示させた。2行目には,入室中の人数および入退室をした者の氏名を表示した(図7)。



図7 LCDディスプレイの画面例

(2) RGBフルカラーLED

1個のLEDで,動作により色を変えるようにしました。

(入力時のカードとの連動)

- ・青色 カード待機中
 - ・緑色 DBに登録済みの
 - 有効なカードが触れた場合点滅
 - ・赤色 無効なカードが触れた場合点滅
- (ボタンとの連動)
- ・青ボタン 水色に点滅
 - ・赤ボタン 10秒間赤色の点滅



図8 各動作によるLEDの発光

2.2.4 Webサイト(管理者のみ)の処理について

入退室の状況や履歴確認をするために,簡単なWebサイトを作成した。このサイトのユーザは閲覧権限のある管理者(先生等)を想定している。

この Web サイトは、次のように構成される。

- ・ トップページ
- ・ ログインページ
- ・ 入退室管理ページ
- ・ ログアウトページ

入退室のログは次（図 8）のように表示される。

The screenshot shows a web application titled "Room Management Service". At the top right is a "ログアウト" button. Below it is a section titled "入退室ログ" with a "件表示" dropdown set to 10. A table lists log entries with columns for "日時" (Date), "名前" (Name), and "入退室時間" (Entry/Exit Time). One entry for "Error Card" at 15:48 is highlighted in red. Below the table is a section titled "入室している人物" (People in the room) with a table showing "IDm", "名前" (Name), and "入室時間" (Entry Time).

図 8 入退室管理ページ

入退室管理ページは、入退室の履歴と現在の入室者のリストを表示させており、DB から [Log] テーブルと [Room] テーブルを表示させている。入退室履歴には検索できる機能などを付けるにあたり、「DataTables」プラグインを実装した。

入退室の履歴は 1 か月前のデータは削除される。また、更新された DB を表示するために、10 秒ごとにページを自動リロードする様にした。

2.3 開発環境

表 1 開発環境

機器	Raspberry Pi 3 model B+
OS	Raspberry Pi OS
使用言語	Python3.7, HTML
使用ライブラリ	Sqlite3, flask, nfcpy
フレームワーク	Bootstrap5, DataTables

3 今回の研究での成果

今回の研究は Python に慣れていないこともあり、自分で作ったクラスファイルのインポートに躊躇ってしまったが、様々な事例などを調べることにより、解決の糸口を見いだすことができた。ま

た、DB やローカルサーバを使用したため、それらを連携させるのが難しかったが、カードの情報を読み込んでサーボモータの動きを管理するなど、システムの制御が思うようにできたので、制御に関しての理解を深めることができた。研究の進行自体は当初の計画より早く進めることができたのによかった。今回は小規模なシステムの製作だったので、もっと大きな IoT システムを作つてみたいと思った。

今回はミニチュアでの研究だったが、実際に使う際には課題がいくつか残されている。一つはサムターン部分のマウント部品をどう固定するかということである。今は金属のおもりで押さえているだけなので、ドアへの固定方法は改めて考える必要がある。もう一つはセキュリティの問題である。学生のいる時間帯（通常の授業時間）は問題ないが、放課後では在室する者がまばらになり、常に鍵を開けたら数秒後に自動で鍵が閉まるようにするなど、セキュリティの工夫が必要であると思う。

まだまだ改善の余地はあるものの、今回の研究を通して組込系システム製作の醍醐味を味わうことができたことは、就職に向け、大きく前進できたと思う。

4 参考文献

- Raspberry Pi でスマートロック作つてみた
<https://qiita.com/undo0530/items/669b6ee6b2277ea64b33>
- Flask と SQLite の連携方法
<https://coffee-blue-mountain.com/python-flask-sqlite3-firstly/>
- Bootstrap と DataTables の連携
https://blog.idcf.jp/entry/bootstrap4_1