非接触 IC カードと Raspberry Pi を用いたシステムの構築

研究者：後藤　太聖・宮川　託実

１　はじめに

Raspberry Pi と非接触 IC カードについて知り理解を深めると同時に、システムを構築することを目的として研究を行った。

本研究ではサーボモータで鍵を制御するシステムと入退出管理システムの２つを構築した。また､非接触 IC カードは FeliCa を用いた。 FeliCa とは、Sony が開発した非接触 IC カードの技術方式及び登録商標の名前である。以下、非接触カードを FeliCa とする。

２　研究内容

本研究では、FeliCa の ID を読み取る機器である PaSoRi とシングルボードコンピュータである Raspberry Pi を用いて、FeliCa の固有 ID 読み取り、サーボモータの制御、入退室のデータベース化に取り組んだ。

（１） FeliCa の読み取り

Raspberry Pi と PaSoRi を USB で接続し、nfcpy PythonModule をインストールして、FeliCa をかざすことで固有 ID を読み取るようにした。

図１　FeliCa を PaSoRi にかざす様子

FeliCa には製造時 IC チップに記録されている書き換え不可能な固有 ID 暗号が存在する。

```
pi@raspberrypi:~ $ cd felicaload.py
pi@raspberrypi:~/felicaload.py $ sudo python felicaload.py
No handlers could be found for logger "nfc.llcp.sec"
Type3Tag 'FeliCa Lite-S (RC-S966)' ID=012E44A8C4962B87 PMM=00
88B4
pi@raspberrypi:~/felicaload.py $
```

図２　ID 読み取り成功画面

図２の下線のように 16 桁の英数字で表されているものが ID である。

（２） サーボモータによる鍵の制御

読み取り成功した ID を、事前に登録されている ID であるかを判断し、一致すれば FeliCa に登録した名前の「Ogaki Taro」がターミナルに表示され、サーボモータが動き、施錠、開錠されるプログラムを Python で作成した。また、読み取った ID が登録したものと違う場合、「Not Authorized」と表示されるようにした。（図３）

```
Locker>>Waiting Card...
No handlers could be found for logger "nfc.llcp.sec"
Locker>>Auth Not Authorized

Locker>>Waiting Card...
No handlers could be found for logger "nfc.llcp.sec"
Locker>>Auth OK: Ogaki.Taro
```

図３　FeliCa を読み取った結果
（上：未登録　下：登録済）

次に、Raspberry Pi でサーボモータを制御させるために Raspberry Pi についている GPIO ピンを用いた。

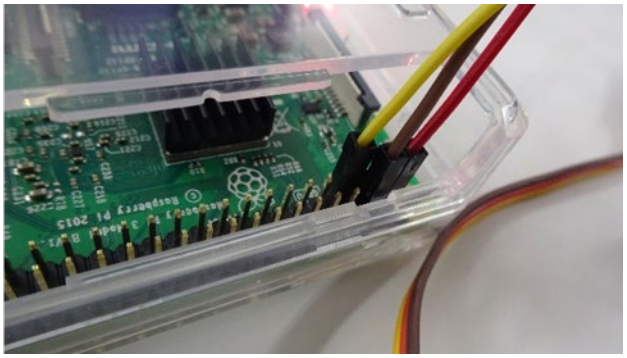


図４　Raspberry Pi の GPIO に接続した様子

（３） SQLite について

SQLite は、小型で処理が速く、使用者一人のみで問題解決ができ、機能性の高いリレーショナルデータベースである。

今回は、Python で SQLite のデータベースを作成した。テーブル内のカラムに、レコードとして、生徒の氏名、ID、時間、生徒の状況（「入室」または「退室」）を格納したデータベースを作成した。

```
import sqlite3

dbname = 'idm_list2.db'
conn = sqlite3.connect(dbname)
cur = conn.cursor()

cur.execute('INSERT INTO Felica(name,IDm,time,status) values("Taro","012E44A8C49655B4","14:26","online")')
cur.execute('INSERT INTO Felica(name,IDm,time,status) values("Hanako","012E44A8C4965F70","15:32","offline")')
conn.commit()

cur.close()
conn.close()
```

図５　SQLite のデータベース作成プログラム

（４） 入退室管理システムの構築

FeliCa を PaSoRi にかざすと、入退室した生徒の氏名、入退室した時間と状況を管理できるプログラムを Python で作成した。このプログラムでは、テーブル内において、事前に登録されている生徒の状況が、「入室」である場合、FeliCa をかざすと「退室」に変更され、「退室」である場合、「入室」に変更される。また、それぞれ入室した時間、退室した時間が記録される。

```
import binascii
import nfc
import signal
import datetime
import sqlite3

class MyCardReader(object):
    def on_connect(self, tag):
        print ("読み取り中")
        self.idm = binascii.hexlify(tag.idm)
        return True
    def read_idm(self):
        clf = nfc.ContactlessFrontend('usb')
        try:
            clf.connect(rdwr = {'on-connect': self.on_connect})
        finally:
            clf.close()

class DatabaseEdit:
    def updatetime(self,idm):
        dt_now = datetime.datetime.now()
        now = dt_now.strftime('%H:%M')
        con = sqlite3.connect('idm_list2.db')
        cur = con.cursor()
        data = cur.execute('select * from Felica')
        data = data.fetchall()
        for row in data:
            print(row[0]+"さん")
            print(now)

        if row[3] == "offline":
                cur.execute('update Felica set time= ?, status= ? where IDm= ?', (now,"online",row[1]))
                print("入室しました。")
                con.commit()
        else:
                cur.execute('update Felica set time= ?,status= ? where IDm= ?' , (now, "offline",row[1]))
                print("退室しました。")
                con.commit()

if __name__ == '__main__':
    cr = MyCardReader()
    de = DatabaseEdit()

    while True:
        print ("カードをタッチしてください")
        cr.read_idm()
        new_idm = str(cr.idm)
        new_idm = new_idm[2:-1]
        print ("カードのID: ",end='')
        print (new_idm)

        de.updatetime(new_idm)
        signal.signal(signal.SIGINT,signal.SIG_DFL)
```

図６　入退室管理システムプログラム

```
pi@raspberrypi:~ $ sudo python3 felica_io0115.py
カードをタッチしてください
読み取り中
カードのID: 012e44a8c49655b4
Taroさん
22:13
退室しました。
カードをタッチしてください
読み取り中
カードのID: 012e44a8c49655b4
Taroさん
22:13
入室しました。
カードをタッチしてください
```

図７　実行画面

３　研究過程

４～６月　：調べ学習
　　Raspberry Pi の初期設定
　　PaSoRi の初期設定
７、８月　：システム作成開始
　　FeliCa の読み取り
　　SQLite のデータベース作成
　　サーボモータのプログラム完成
９、１０月　：FeliCa 判別プログラム作成
　　入退室管理システムの構築
　　展示用ドアの製作
１１月　：文化祭展示
　　文化祭ポスター作成
１２月　：レポート作成開始
１月　：資料の作成と発表

４　研究成果

（１）　PaSoRi で FeliCa の読み取り

Raspberry Pi での初期設定や Python、nfcpy などのインストールの方法をインターネットで調べ参考にすることで、FeliCa の読み取りに成功した。自力でこのプログラムを作成するのは大変難しいことだということが分かった。

（２）　Raspberry Pi でサーボモータの制御

Raspberry Pi の GPIO ピンには、それぞれに役割があり、理解することができた。また、モーターに関する知識も学ぶことができた。今回の制御では、パルス幅の高さを変える PWM と、パルス信号のオンオフの比率を変更するデューティー比の２つを理解し制御できた。

（３）　SQLite のデータベースの制御

テーブルに格納されているデータを更新するために、Python のプログラム内で SQL の update 文を用いた。結果、テーブルが更新され、入退室管理をすることができた。

５　課題

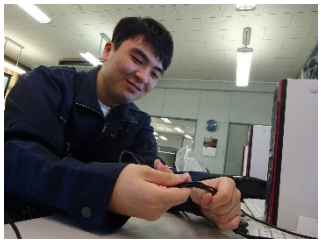
サーボモータによる鍵の制御での課題は、まだ実用的なレベルではないというところである。スマートフォンなどの他種デバイスとの連動を実現するべきだ。

また、SQLite のデータベース作成の課題は、現在のデータベースの表示画面は少し見にくいというところである。そこを修正して見やすくするべきだ。

６　チームの感想と反省

【後藤　太聖】

4～6 月の間は Raspberry Pi と FeliCa について知識を深め、自分のやってみたい研究内容を探した。その時に、Raspberry Pi にはできることが数多くあるということを知り、その時はどんなものを作ろうかと少し楽しみながら考えていたが、Raspberry Pi 内でシステムを作るには Python が欠かせないということを知り、言語の習得に対して少々不安が生まれた。ところが、構文が意外とシンプルで分かりやすく、プログラムを完成させることができた。本研究でできた経験は良いものだと思った。



【宮川　託実】

Python や SQLite を初めて使用したため、いくつものエラーメッセージの解読に苦戦したが、調べ学習を行ったことや担当の先生に、わからないことを教えていただき、多くの問題点を解決しながら研究を進めることができた。また、SQLite のデータベースの作成は非常に簡易ではあったが、Python との接続が上手くできず少々時間を費やしすぎてしまったため、さらに、Python と SQLite について学ばなければならないと実感した。本研究では、FeliCa と Raspberry Pi の仕組みなどを学び、理解を深めることができ、研究で得たことを今後の人生でも生かしていきたいと思った。