

# IC カードを使った入退出記録装置

## Entrance/ exit record device using IC card

芝浦工業大学 無線研究部

Shibaura Institute of Technology, Ham radio club

### 1. 動機

もともとモノとインターネットを通じて生活を豊かにする Internet of Things(IoT)に興味があり、この機会にそれに関連するモノを作りたいと思った。そこで大学施設にある IC カードを使った入退出記録装置に目を付け、同じようなものを作り日常生活の質の向上に貢献できないかと考えた。

### 2. 目的

IC カードを用いた入退出記録装置を作製し、生活の質を上げるとともに、自身の学習に役立てる。装置は実用に耐えることを目標とする。

### 3. 製作

環境

PC : Raspberry Pi 3 model B

OS : Raspberry Pi OS(10.0)

言語 : Python3.9.6

- ・ユーザを登録するプログラムの作成

個人の入退出を記録するために、まずカードとユーザを紐づける必要がある。今回は、users\_list.txt 内で同じ行にカードの固有 ID とユーザ名を記録することでこれを実現した。

- ・カードを読み込み入室・退室時刻を記録するプログラムの作成

カードをタッチし、上のユーザ登録プログラムで登録したユーザの入室・退室時間を記録する。ユーザ名の取り出しは以下のように行う。

カードをタッチする→カードの固有 ID を読み込む→users\_list.txt 内でカードの固有 ID と一致する行を検索する→一致する行のユーザ名部分を読み込む

上の仕組みを使ってカードをタッチした際に登録されたユーザを呼び出し、ユーザ名.csv (例: 芝浦太郎.csv) に入室・退室の状態、タッチした時刻、入室から退室または退室から入室までの経過時間を記録する。

また、入室・退室どちらとして記録するかは次に行う電子部品との連携によって切り替えられるようにする。

- ・カード読み込みプログラムと電子部品を連携させる

入退出記録装置として使う場合、画面出力を使用しない方がコンパクトになる。そのため HDMI 出力を行わずに入退出記録がスムーズに行えるよう、カード読

み込みプログラムと電子部品による音や光による通知を行う。

今回は①IC カードを読み込める状態であることを知らせる LED、②入室・退室のどちらであるかを知らせる LED、③入室・退室の状態を切り替えるスイッチ、④シャットダウン・起動ができるスイッチ、⑤プログラムの動作をより分かりやすくするための電子ブザーを用意した。

### 4. 結果

IC カードを使った入退出記録ができるようになった。また HDMI による画面出力を行わずともカードを読み込んでいることが分かるようになった。

### 5. 考察

今現在の仕組みでは、入退出記録を確認するために結局 HDMI による画面出力を行わなければならない。また、ファイルに記録をするのみでは個人が実用するうえで扱いづらい部分がある。

### 6. まとめ・展望

最初の目的である「自身の学習に役立てる」について、python の利点を理解しながらプログラミングをし、初めて自分で実際に使うプログラムをかくことができた。電子工作に関しては簡単なものではあるが、実際にブレッドボードを使って LED を光らせ、電子ブザーを鳴らすことができ、自身の学習に役立てることができていると考える。

現状では、入退出記録の仕組みを作ることができたが、電子回路はブレッドボード上であり、また考察で述べたように、入退出記録の確認に画面出力が必要であり、また記録のみでは実用性が低いことが考えられた。

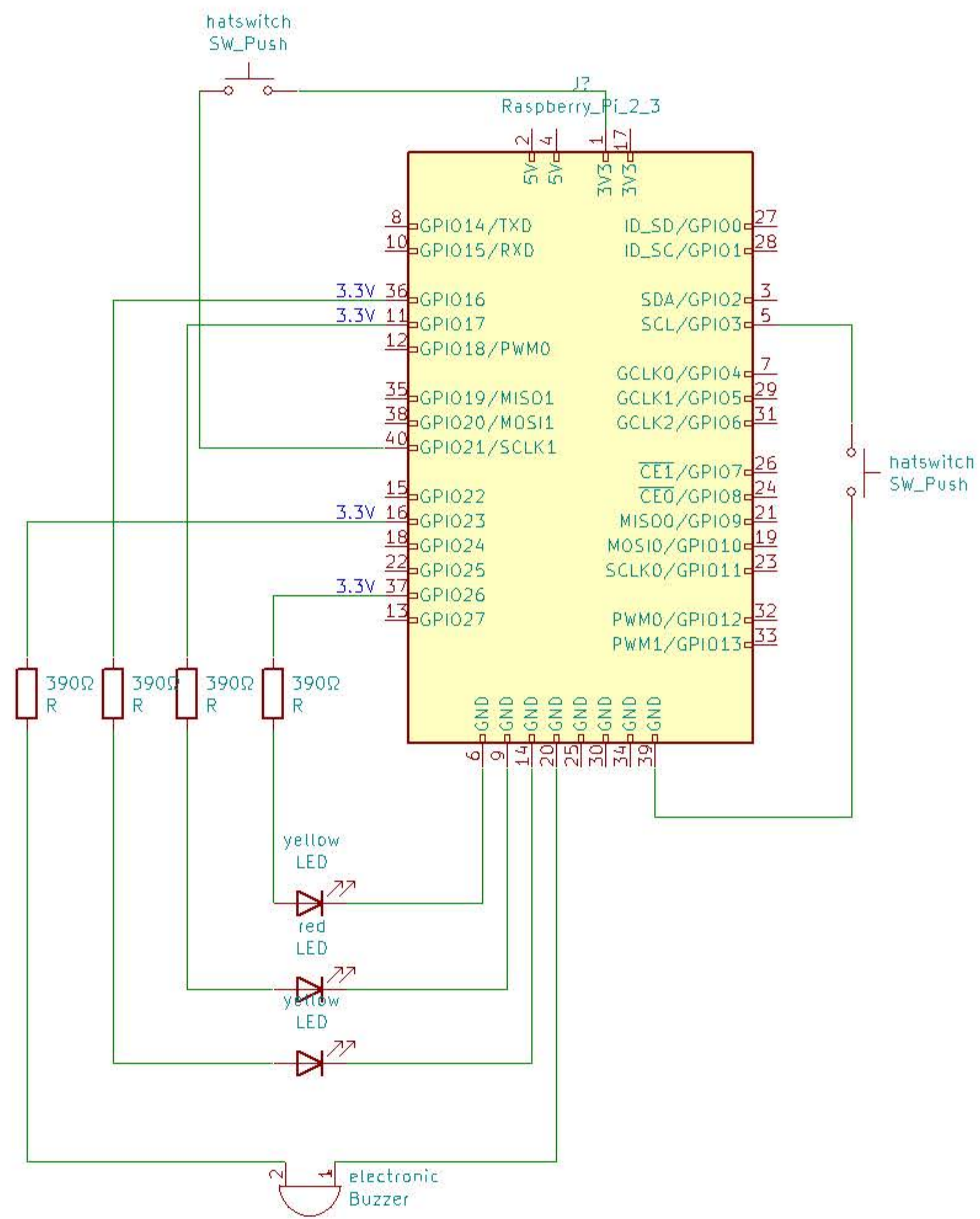
そのため今後は電子回路をユニバーサル基板上で実現し、装置として使えるようコンパクトにまとめる、カードをタッチした際に LINE に通知するようにする、入退出記録ファイルを外部からアクセスできるようにする、などの機能を実装していきたい。

### 7. 参考文献

・Raspberry Pi と NFC タグで自宅の消耗品管理の仕組みを作った話

<https://www.membersedge.co.jp/blog/raspberry-pi%E3%81%A8nfc%E3%82%BF%E3%82%B0%E3%81%A7%E8%87%AA%E5%AE%85%E3%81%AE%E6%B6%88%E8%80%97%E5%93%81%E7%AE%A1%E7%90%86%E3%81%AE%E4%BB%95%E7%B5%84%E3%81%BF%E3%82%92%E4%BD%9C%E3%81%A3%E3%81%9F/>  
最終閲覧日 2021/9/3

• Getting started— nfcpy 1.0.3 documentation  
<https://nfcpy.readthedocs.io/en/latest/topics/get-started.html#read-and-write-tags>  
最終閲覧日 2021/9/3



Sheet: /  
File: read.py.sch

**Title: read.py**

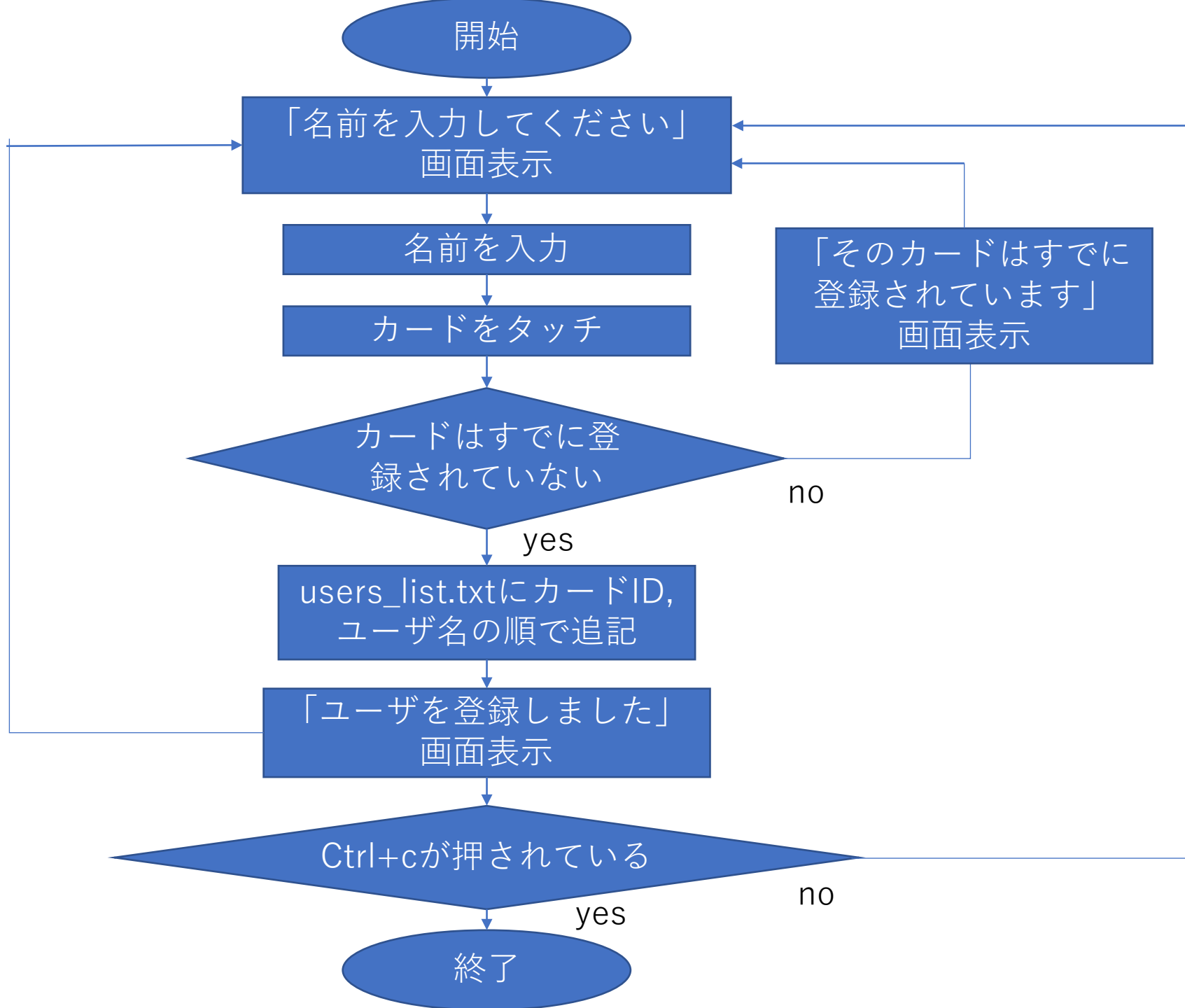
Size: A4 Date: 2021-09-10

KiCad E.D.A. kicad (5.1.10)-1

**Rev:**

Id: 1/1

add\_users.py



```

1 import nfc
2 import csv
3 import binascii
4 import os
5
6 #ユーザを追加する関数
7 def add_users(IDm,name):
8     with open('users_list.txt','a') as f:
9         f.write(str(IDm)+' '+ name + '\n')
10        print("ユーザーを追加しました")
11
12 #ユーザとIDがファイルに存在するか判定する関数
13 def check_func(IDm,name,mode):
14     #ファイルが存在しない場合、Falseを返す
15     if not os.path.isfile('users_list.txt'):
16         return False
17     #存在する場合、IDまたはuserの存在を判定する
18     with open('users_list.txt','r') as f:
19         csv_header = ['card ID','name']
20         for info in csv.DictReader(f,csv_header):
21             #print(info)
22             if mode == 'ID':
23                 if ('card ID',IDm) in info.items():
24                     return True
25             elif mode == 'user':
26                 if ('name',name) in info.items():
27                     return True
28         return False
29 #カードをタッチすると呼び出される関数
30 def on_connect(tag):
31     idm = binascii.hexlify(tag._nfcid)
32     global id
33     id = str(idm)
34     print("IDm : " + str(idm))
35     print('カードを離して下さい')
36     return True
37 #タッチしたカードを離すと呼び出される関数
38 def on_release(tag):
39     return True
40 #ICカードを読み込んだ時に呼び出す関数
41 rdwr_options={'on-connect' : on_connect,'on-release' : on_release}
42 try:
43     while 1:
44         name=input("名前を入力して下さい -> ")
45         print("カードをタッチして下さい...")
46         clf = nfc.ContactlessFrontend('usb:054c:06c3')
47         clf.connect(rdwr=rdwr_options)
48         if check_func(id,name,'ID'):
49             print('そのカードはすでに登録されています')
50         elif not check_func(id,name,'ID'):
51             add_users(id,name)
52         clf.close()
53 #ctrl+cが押された時の処理
54 except KeyboardInterrupt:
55     print('終了')
56

```



```

1 import nfc
2 import csv
3 import binascii
4 import datetime
5 import os
6 import sys
7 import signal
8 from time import sleep
9 import RPi.GPIO as GPIO
10 #statusに初期値として'in'を格納しておく
11 status = 'in'
12
13 #電子ブザーを鳴らす関数
14 def BZ_control(mode):
15     #0.1秒間鳴らす
16     if mode == 0:
17         GPIO.output(23,GPIO.HIGH)
18         sleep(0.1)
19         GPIO.output(23,GPIO.LOW)
20     #0.5秒間鳴らす
21     elif mode == 1:
22         GPIO.output(23,GPIO.HIGH)
23         sleep(0.5)
24         GPIO.output(23,GPIO.LOW)
25
26 #ctrl+cが入力された時の処理
27 def handler(signal,frame):
28     print('終了')
29     GPIO.cleanup()
30     sys.exit(0)
31 signal.signal(signal.SIGINT,handler)
32
33 #ICカードと通信した時に呼び出される関数
34 def on_connect(tag):
35     BZ_control(0)
36     GPIO.output(17,GPIO.LOW)
37     #カードの固有IDを読み込んで代入
38     idm = binascii.hexlify(tag._nfcid)
39     global id
40     #IDmを文字列としてグローバル変数idに代入
41     id = str(idm)
42     print("IDm : " + str(idm))
43     print('カードを離して下さい')
44     return True
45
46 #カードが離された時に呼び出される関数
47 def on_release(tag):
48     return True
49
50 #タクトスイッチが押された時にinとoutを切り替える関数
51 def switch_status(channel):
52     LED_control(0)
53     global status
54     if status == 'in':
55         GPIO.output(26,GPIO.LOW)
56         GPIO.output(16,GPIO.HIGH)
57         status = 'out'
58         print(status)
59
60     else:
61         GPIO.output(16,GPIO.LOW)
62         GPIO.output(26,GPIO.HIGH)
63         status = 'in'
64         print(status)
65
66 #ファイルにIDmと名前が存在しているか判定する関数
67 def check_func(IDm,name):
68     if not os.path.isfile('users_list.txt'):
69         return False
70     with open('users_list.txt','r') as f:
71         reader = csv.reader(f)
72         for line in reader:
73             if IDm in line or name in line:
74                 return True
75     return False
76
77 #in->outの場合に経過時間を計算する関数
78 def calc_staytime(dt_now):
79     with open(f'users/{name}.csv','r') as f:
80         lst = list(csv.reader(f))
81         leng = len(lst) - 1
82         if leng == -1:
83             return
84         if status == 'in':
85             if lst[leng][0] == 'out':
86                 tdatetime = datetime.datetime.strptime(lst[leng][1], '%Y-%m-%d %H:%M:%S.%f')
87                 td = dt_now - tdatetime
88                 #print(f'{dt_now} - {lst[leng][0]}')
89                 print(td)
90                 return td
91
92         elif status == 'out':
93             if lst[leng][0] == 'in':
94                 tdatetime = datetime.datetime.strptime(lst[leng][1], '%Y-%m-%d %H:%M:%S.%f')
95                 td = dt_now - tdatetime
96                 #print(f'{dt_now} - {lst[leng][0]}')
97                 print(td)
98                 return td
99
100 #ユーザー名.csvにstatus,現在時刻,経過時間の順に書き込む関数
101 def write_csv(name):
102     dt_now = datetime.datetime.now()
103     td = calc_staytime(dt_now)
104     with open(f'users/{name}.csv','a') as f:
105         writer = csv.writer(f)
106         print(dt_now)
107         writer.writerow([status,dt_now,td])
108
109 #ファイルを検索してIDmからユーザー名を取り出す関数
110 def display_user():
111     with open('users_list.txt','r') as fr:
112         list_reader = csv.reader(fr)
113         for line in list_reader:
114             if id in line:
115                 name = line[1]
116                 print(f'user:{name}')
117                 break
118     return name
119
120 #GPIOピンセットアップ
121 GPIO.setmode(GPIO.BCM)
122 GPIO.setup(23,GPIO.OUT)
123 GPIO.setup(17,GPIO.OUT)
124 GPIO.setup(21,GPIO.IN,pull_up_down = GPIO.PUD_DOWN)
125 GPIO.setup(16,GPIO.OUT,initial = GPIO.LOW)
126 GPIO.setup(26,GPIO.OUT,initial = GPIO.HIGH)
127 GPIO.add_event_detect(21,GPIO.RISING,callback = switch_status,bouncetime=500)
128 #ICカードをタッチするときに呼び出す関数の指定
129 rdwr_options={'on-connect' : on_connect,'on-release' : on_release}
130
131 while 1:
132     print("カードをタッチして下さい...")
133     GPIO.output(17,GPIO.HIGH)
134     global clf
135     #カードリーダーを探す
136     clf = nfc.ContactlessFrontend('usb:054c:08c3')
137     #ICカードと通信する
138     clf.connect(rdwr=rdwr_options)
139     clf.close()
140     #idがファイルに存在するか
141     if check_func(id,''):
142         #存在する場合、idmに対応するユーザー名を取得
143         name = display_user()
144         #ユーザー名.csvの存在確認、存在する場合、書き込む
145         if os.path.isfile(f'users/{name}.csv'):
146             write_csv(name)
147         #存在しない場合、ヘッダーを書き込みその後通常通り書き込む
148     else:
149         f = open(f'users/{name}.csv','w')
150         writer = csv.writer(f)
151         writer.writerow(['status','time','elapsed time'])
152         f.close()
153         write_csv(name)
154     #存在しない場合の処理
155     else:
156         BZ_control(1)
157         print('登録されていないカードです')

```