

フリーソフトウェアを用いたRPAによる業務自動化について

常三島技術部門
地域協働グループ

飯田 仁(HIDA Hitoshi)

Keywords : RPA, SikuliX, MIFARE, 学生証

1. はじめに

近年、RPA(Robotic Process Automation)を活用することにより、業務の自動化を行い、作業負担の軽減が活発に推進されている。徳島大学においても、複数の有料RPAシステムが導入され活用されている。予算の関係で有料RPAシステムを利用できずにいたが、今回、フリーソフトウェアを用いたRPAにより業務を自動化することができたので報告する。

2. 作業の概要

徳島大学では休祝日及び夜間の建物への入館は大手警備会社(以下S社)の電気錠システムを導入し、非接触式ICカードである職員証と学生証を入館鍵として使用し、許可された者だけが建物に入館することができる。

何らかの事情で学生証を再発行した場合に受け取る新学生証には、徳島大学の授業出席管理システムで利用するデータは既書き込まれているが、入館鍵として使用するため以下の2つの作業が必要である。

- ① S社の入館鍵データを新学生証に追書き込みする。
- ② 入館に関する許可情報を管理している電気錠システムサーバに追書き込みしたデータを登録し、学生証と電気錠システムサーバのデータ同期を行う。

学生証に追書き込みする入館鍵データには発行回数という情報が含まれており、電気錠システムサーバでこの発行回数を適切に更新することで、紛失した学生証を拾得・利用しても入館することができないようにしている。

3. 作業内容

① 学生証へのデータ追書き込み

新学生証への入館鍵データの追書き込みは、S社の専用書き込み装置(図1)と専用書

き込みソフトウェア(図2)にて実施する。入館鍵データの内容はグループコード・個人コード・発行回数の3種類で、グループコードと個人コードは学生番号から求めることができる。発行回数は新学生証表裏面には記載されていないが、データとして記録されている。しかし、ここでは読み取ることができないため、筆者は次で説明する電気錠システムサーバのデータを参照し、現在値に1加算した値を使用した。

専用書き込みソフトウェアに、グループコード・個人コード・発行回数の値を入力し、書き込みボタンをクリックすれば、追書き込みの作業は完了である。



図1 専用書き込み装置の外觀

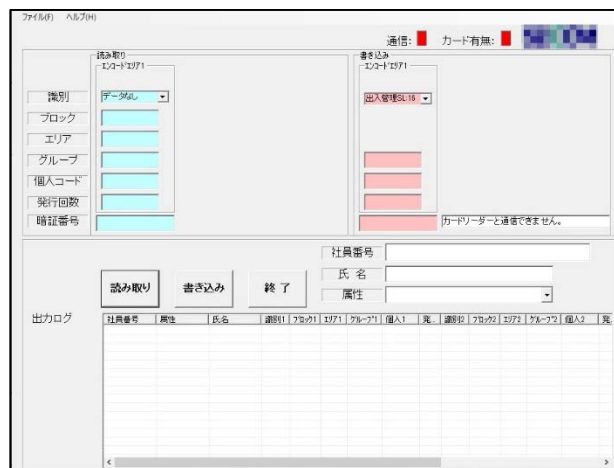


図2 専用書き込みソフトウェア画面

② 電気錠システムサーバデータの同期処理

新学生証への追書き込み作業が終われば、電気錠システムサーバのデータと新学生証のデータが一致しない（前述の発行回数）ので、このままでは入館資格を持っていても新学生証では入館することができないため、電気錠システムサーバ上の発行回数データを修正する必要がある。この作業はWebブラウザにて実施する。電気錠システムサーバにログインした後の手順を以下に示す。

- イ) 当該学生のデータを検索・表示させ一人分のデータがあるcsvファイルをダウンロードする。
- ロ) ダウンロードしたcsvファイルの発行回数データを、新学生証に追書き込みした値に変更し保存する。
- ハ) 発行回数を変更したcsvファイルをアップロードする。

以上の操作で実際に新学生証に追書き込みしたデータと、電気錠システムサーバのデータとの同期をとることができる。ただし、ハ)の作業についてはほぼ全学の電気錠システムを1台のサーバで管理しているため、別の担当者が作業中の場合、1回で正常にアップロードできず、何度も同じ処理を繰り返す必要があり、時間を取られることもあった。

4. RPA化の作業内容

RPA化する作業は、2で示した作業全部であるが、2-①で示した発行回数の確認は後述の読み取り装置を新たに作製したため新学生証に書き込まれている発行回数を利用することとし、筆者の実作業を変更・簡略化した。

5. RPAシステム概要

システムは小型PCを使用し、新学生証から情報を読み取る装置は過去に製作した出席管理システム^[1]のカードリーダーを改良・流用し、新学生証への書き込み装置類はS社のカードライター（H50×W120×D220mm）を無加工で利用した。システムを中心となるRPAにはフリーソフトウェアのSikuliX^[2]を利用した。SikuliXについての詳細は割愛するが、プログラムによりパソコン上の様々な作業を自動化することが可能である。システムの概要

を図3に示す。今回作製したシステムは、H100×W270×D360mmの小型の筐体に収納することができ、事務机や、カウンター上に設置可能な大きさに仕上げることができた（図4）。徳大カードリーダーとS社カードライターは3mmの学生証挿入スペースを設けた上で、お互いの電波が干渉しない様に、同時に電源が入らない仕様とした。

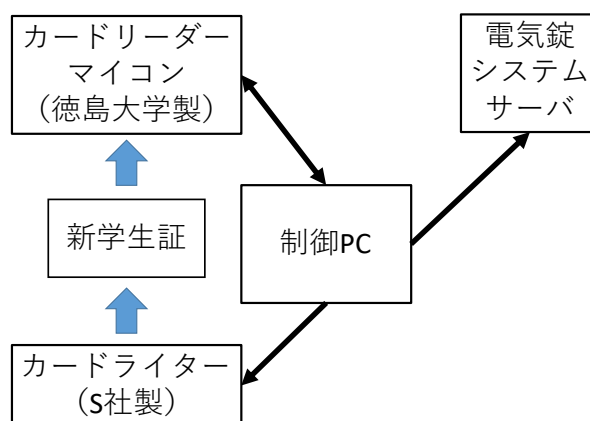


図3 システム概要



図4 完成したシステム外観

5-1 カードリーダー部分

カードリーダー部分は、出席管理用カードリーダーを改良したもので、カードリーダーモジュールをマイコンで制御し新学生証から学生番号と発行回数を取得、文字列として制御PCに送信する。また、制御PCからの指令に基づき、追書き込みが終わった新学生証を筐体から排出する制御も行う。カード排出機能は、ソレノイドのONにより実現し、ON時間を調整することで新学生証の不必要な飛び出しを抑制した。制御PCとマイコン間はRS232Cのシリアル通信でUSB-シリアル変換ICケーブルを用いた。

5-2 カードライター部分

カードライター部分は、S社製のカードライターを無加工で利用し筐体内に収めた。本来ならば筐体内に収めるため、カバー類の取り外しと電源用ACアダプターの小型組み込み電源への交換をしたかったが、S社からの苦情（クレーム）の可能性を考え一切手を加えなかった。制御PCとカードライター間もRS232Cのシリアル通信でUSB-シリアル変換ケーブルを用いた。

5-3 制御PC部分

制御PCとしてWindows10が動作する小型PCであるLattePanda Alphaを使用し、小型7インチタッチパネルとNVMeSSDを追加した。この制御PCに、カードライター制御ソフトウェア（S社製）（図2）と、カードリーダーを制御するソフトウェア（本学製）（図5）を導入した。さらに本原稿で最も重要なSikuliXを導入した。SikuliXはJavaの開発環境も必要になることから、無料で利用できるJava開発環境JDKも導入した。今回、SikuliXのプログラムはPythonを用いて開発したがPythonの環境はSikuliXと同時に導入されている。

SikuliXはPC画面を画像認識しマウス移動やキーボード入力といった操作を実行可能である。適切に処理をプログラムすることで今回のようにRPAとして利用可能である。筆者は主にC言語にてプログラム開発を行っており、Pythonの使用経験はなく戸惑うことが多かったが、SikuliXに関する良書^[3]を書店で偶然見つけることができ非常に役立った。さらに、Pythonについてはネット上に多くの情報があり、参考にすることで比較的短時間で動作させることができた。しかし、完成の域に達するまでは試行錯誤が必要で時間を要した。



図5 カードリーダー画面

自動化処理の流れとして①の動作は

- A) 図4に示す装置前面のカードスロットに学生証を差し込むことで、マイコンが学生証から必要な情報を読み取り、制御PCに送信する。
- B) マイコンから送られた情報を基に、図5に示すカードリーダー画面左側に学生番号・cアカウント・グループコード・個人コード・発行回数を表示する。さらに、SikuliXの自動動作開始トリガとなるダイアログボックスを表示する（図6）。
- C) SikuliXが図5に示す画面のグループコード・個人コード・発行回数をコピーし、図2に示すS社専用書き込みソフトウェア画面右上の、グループコード・個人コード・発行回数の入力欄にそれぞれをペーストする。
- D) 正常にペーストされたことを確認し図2画面中央左の『書き込みボタン』をクリックする。
- E) 図5に示す徳大カードリーダーソフトウェア画面に戻り、左下の『書き込み完了ボタン』をクリックすると同時に、マイコンに対し学生証排出の指令を送る。
- F) 装置から不必要な飛び出しなく学生証が排出される。

以上が、正常に書き込みが行われた際の動作であり、当然書き込みできない場合もあるので、それらの処理も不具合発生ごと適切にプログラムした。A)はマイコンの処理。B)は徳大カードリーダーソフトウェアの処理。C)～E)がSikuliXによる自動化（RPA）処理。F)はマイコンの処理である。これらの処理中は、マウスが自動で移動し様々な動作を行うため、マウス操作はしてはならない。過ってマウスを動かした場合は、自動処理が停止してしまうので注意が必要であり、筆者はプログラム開発中に何度も経験している。本番環境ではマウス・キーボードは接続しないので気にする必要はない。

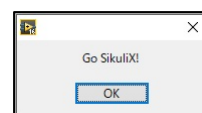


図6 SikuliX起動用ダイアログボックス

次に、自動化処理の流れとして②の動作は、毎日21時にバッチ処理として実施する。

- a) 当日にカードライターの書き込みが行われていたら以降の処理を実施し、書き込みが行われていない場合は処理終了。判断は、S社書き込みソフトウェアが自動的に書き込みデータの記録用csvファイルをlist_yymmdd.csvという日付入りのファイル名で作成するので、内容を読み取ることで行う。
- b) 書き込みが行われていれば、Webブラウザを起動し、電気錠システムサーバに接続する。
- c) 認証のため、ユーザー名・パスワードを画面上の必要箇所に入力し、ログオンボタンをクリックする(図7)。
- d) 画面上のユーザー管理タブをクリックし、メニューを表示後、ユーザー管理をクリックする(図8)。
- e) 画面上のcsv出力ボタンをクリックする(図9)。
- f) ダイアログボックスが表示されるので全データをクリックし、登録されている全データをダウンロードする(図10)。
- g) 確認用ダイアログボックスが表示されるので、はいをクリックする(図11)。
- h) ダウンロード(DL)完了のダイアログボックスが表示されるので、ダウンロードしたcsvファイルのファイル名をダブルクリックで選択、コピーし完了をクリックする(図12)。
- i) b)で書き込んだ新学生証の内容と、h)でダウンロードした内容を比較し、新学生証のデータのみを抽出、発行回数を修正した更新用csvファイルを作成する。比較作業にh)でコピーしたファイル名を用いる。
- j) 図9の画面右下のcsv入力ボタンをクリックし、ファイル選択画面から更新用ファイルを指定し、開くボタンをクリックする。
- k) いくつかのダイアログボックスが表示されるので全てOKボタンをクリックし、最終的に「処理が正常に終了しました。」というダイアログボックスを確認し閉じる

ボタンをクリックする(図13)。

- l) ブラウザを終了する。

以上が、正常に更新できた際の最短動作であり、当然更新できない場合もあるので、それらの処理も適切にプログラムした。a)~h)とj)~l)がSikuliXによる自動化処理。i)はSikuliX内の処理ではあるがPythonによるファイル処理である。



図7 ログイン画面



図8 ユーザー管理選択画面



図9 csvボタン画面



図10 データ選択ダイアログボックス



図11 確認用ダイアログボックス

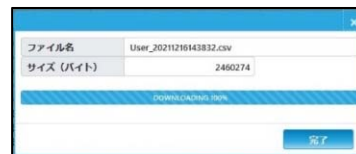


図12 DL完了ダイアログボックス



図13 終了ダイアログボックス

6. SikuliXのプログラム紹介

今回利用したPythonで記述したSikuliXのプ

プログラムを図14に紹介する。通常Pythonのプログラムはテキストのみで記述されるが、SikuliXでは画像を取り込み利用できるため、開発当初は少し違和感があった。

図14はS社のライターソフトウェアが起動しているかを確認する処理として実際に使用しているプログラム例である。文字列の部分はおなじみのIF文で条件分岐に利用するが、その次に図2に示す画像の一部（中央左側付近）の3つのボタン画像が来ている。ボタン画像に続く0.2は確認する待ち時間で秒を入力する。この場合は『ボタン画像が画面上に0.2秒間待っても表示されていなければ』TRUEとなり、ライターソフトウェアの起動処理が実行される。

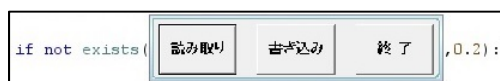


図14 SikuliXのプログラム例1

7. 学生証再発行時の流れ

説明が前後するが、筆者の所属（というか派遣されている）している理工学部には、現在8コースあり、学生が学生証を再発行する場合には、同じ常三島キャンパス内の学務部教育支援課教務・情報係（以下、発行担当）に必要書類を提出し、手続きを行う。

必要書類を受理した発行担当では学生証発行機を利用して、学生証の表面（写真面）の印刷と、徳島大学の教務システムで利用するデータを書き込むことで新学生証を再発行し学生に渡している。旧学生証を持っている場合は交換となる。印刷にも時間が掛かるので再発行には数日必要となる。

新学生証を受け取った学生は、自身が所属しているコース（学科）の担当者の所に出向くように発行担当から指示され、今回の作業を各コース担当が実施している。各コース担当が不在の場合は、何度か出向く必要があり、学生にとって負担となっている。

ここまで書くと何故『学生証発行機を持っているならば、学生証表面の印刷・発行時に徳島大学で利用する情報に加え、S社の入館鍵情報も同時に書き込まないのか？』と疑問に思う方も居ると思う。筆者は直接確認して

いないが、発行担当もS社と協議をしているようだが、S社の入館鍵情報の書き込みに関する『ICカードの書き込み鍵』情報が開示してもらえないため、同時書き込みができない状態である。

今回のRPAを用いた自動化の一部（①）については、以上のような根本原因を解決すれば不要であったとも考えることができ、学内では解決できないが、現在の業務方法が業務効率改善の足枷となっている典型例であると筆者は感じている。

8. 今後の展開

現状は、いまだ確認状態ではあるが、学生証発行装置のある場所に設置し、再発行後速やかに本システムを利用し必要情報を全て書き込んだ新学生証を学生に渡すことで、学生に加え、筆者などの各コース担当者の負担も軽減できると考えるので、発行担当と交渉していく。また、一部の情報では次年度から学生証の規格が現状のMifareからFeliCaに変更するとの話もあるが、RPAプログラムとS社製カードライターを変更すれば対応でき、ベルト駆動装置等の追加により自動で新学生証を移動・交換することで、現在外注化している、学部・大学院合わせて毎年約2,000人の新入生の学生証発行業務にも対応可能である。

9. まとめ

原稿執筆時点では、システムは確認状態であるが、フリーソフトウェアを利用したRPAが実現できた。手作業の手順を今回のカードリーダーを利用するなどように少し変更することで、RPAの処理が簡単になる場合もある。

参考文献

- [1] 飯田 仁, 吉永 哲也, “学生証を用いた授業出席管理システムの開発”, 大学教育研究ジャーナル 第7号 2009年3月 pp.131-137 徳島大学
- [2] <http://sikulix.com/>
- [3] さわって学べるSikuliX/Pythonで作るRPA 大沢文隆 日経BP