

LabVIEWによる在館管理システムの開発

東島, 亜紀
九州大学応用力学研究所

<https://doi.org/10.15017/4794813>

出版情報 : 九州大学応用力学研究所技術室 技術室報告. 4, pp.43-47, 2022-07. Research Institute for Applied Mechanics, Kyushu University

バージョン :

権利関係 :

LabVIEW による在館管理システムの開発

東島 亜紀

要 旨

高温プラズマ理工学研究センターのクエスト実験棟は、大型実験装置や周辺設備・機器等が設置された比較的大きな建物である。教職員および学生、関係者の出入りも多く、電話等の取次、実験棟内の消灯や施錠、災害時における退避確認などのために、在館者を常時明らかにしておきたい。今回、実験棟の入退出（在館者）状況を表示する「在館管理システム」を、低コストおよび運用管理作業負担減を目指し、LabVIEW 等を用いて開発した。

キーワード

在館管理 システム開発 LabVIEW Flask Python

1. はじめに

高温プラズマ理工学研究センターのクエスト実験棟は、大型実験装置および周辺設備・機器がある本体室や電源室、その他、工作室や実験準備室、制御室や会議室等を内包する比較的大きな建物である。1階廊下には、各人専用のランプ付き押しボタンスイッチにより実験棟内の在籍を切り替える盤（以下、切替盤）があり（図1）、該当者の在籍を確認できる表示盤が2階に2か所設置されている（図2）。長年、これらの在籍盤により在館者の確認を行い、実験棟内の施錠や消灯、電話や来訪者の取次などに対応してきた。しかし、1階廊下の入口ドアが、セキュリティ向上を目的とした認証式となり、在籍の切り替えに「ロックを解除してドアを開ける」という一手間が増えた。そのため、次第にスイッチを押す者が少なくなり、これらの在籍盤は形骸化していった。

消灯や施錠、取次等の確認、また、災害時において在館者が棟外に出て避難が完了したかどうかの確認は特に重要であり、実験棟への入退出を確認できる在籍盤が形骸化したままでは問題がある。しかし、切替盤側の移動は配線等の問題で難しく、また、掲示メンバー数の増加や氏名変更など運用面における柔軟性も乏しいため、在籍盤の再活用には消極的であった。そのため、実験棟への入退出を管理できる新たな「在館管理システム」を玄関ホールに導入する要望が高まってきた。



図1 押しボタンによる切替盤

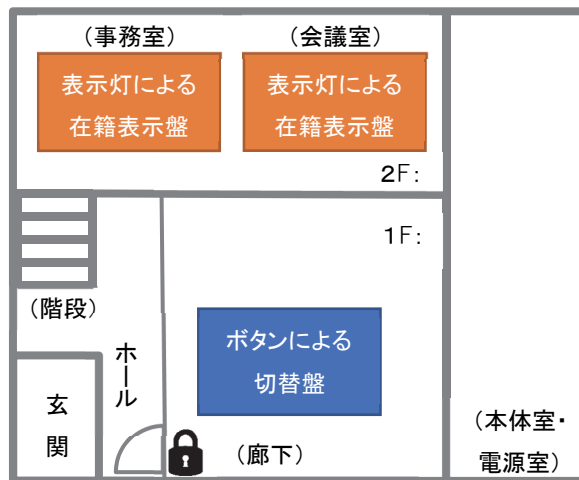


図2 実験棟内の在籍盤

2. 在館管理システム開発の背景

共同利用研究者など常時在館しない関係者の来訪も多い実験棟であるため、ID カードによる認証式入退管理システムなどの既製品導入は、コストや運用管理の手間を考えると難しい。また、掲示メンバーの数や氏名の変更、機能追加など運用面での柔軟性に不安がある。

高温プラズマ理工学研究センターでは、GUI 作成が簡単に行えるグラフィカルプログラミング言語 LabVIEW を導入していることもあり、クエスト実験棟に適した在館管理システムを独自開発することとなった。

3. 在館管理システムの要件と構成

「ボタンを押す」操作のみの在籍盤の運用は、長年の実績がある。在館管理システムは、この手軽さを引き継ぎ、LabVIEW プログラムで対応する。在館切り替え操作を行う側（以下、在館操作側）は、タッチパネル端末上で動作させ、在館者表示側との情報伝達は、直接配線からネットワークへと変更する。また、在館操作および在館者表示が出来るという主要件以外に、次のような要件があがった。

- 掲示メンバーの変更は、在館操作側・在館者表示側において一括で行えること。
- タイムカードの代用にもなるように入退出時刻を記録する機能を追加すること。
- 在館者表示側は必要に応じて増設できるようにすること。

これら要件を満たし、在館操作側と在館者表示側で必要な情報伝達を検討した上で、図3のようなシステム構成とした。

このシステムは、在館操作側、在館者表示側の2つの LabVIEW プログラムを常時実行させ、Status ファイルを介し、リアルタイムの在館者表示を行う。設定ファイルは、掲示するメンバーの基本情報を持ち（表1）、Status ファイルは、最新の各メンバーの在館状況などの情報を持つ（表2）。また、設定ファイルで Time-Record を ON にしたメンバー分のみ、在館操作した時刻を、個人の Time-Card ファイルに記録する。各ファイルは、既存の内部サーバへ、プログラム側でアップロード・ダウンロードを行う。

在館操作側、在館者表示側どちらも、LabVIEW がインストールされた端末ではなくても実行で

きるように、開発後のプログラムは EXE 化する（端末には、LabVIEW ランタイムのインストールが必要）。

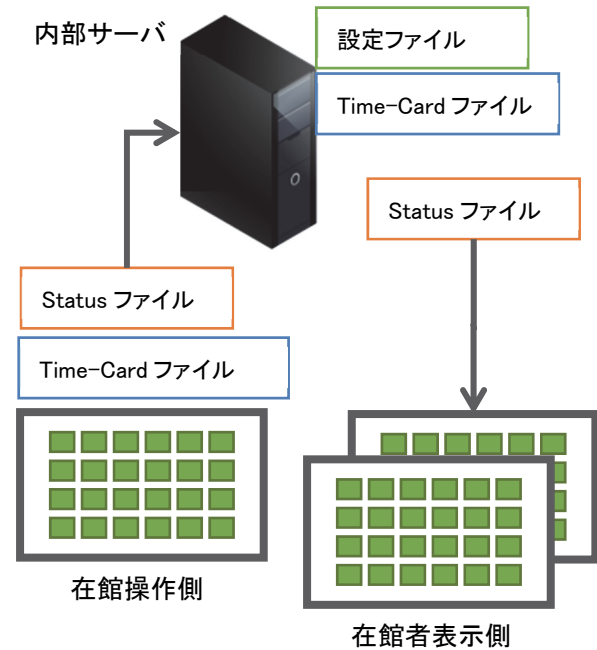


図3 在館管理システム構成

表1 設定ファイルの項目

Name	Disp.Name	Time-Record ON/OFF
東島亜紀	東島	ON
...

表2 Status ファイルの項目

Name	Disp. Name	Status T/F	In_ Time	Out_ Time
東島亜紀	東島	T	2022/ 04/...	
...

表3 在館操作側の機器構成

タッチパネル	I・O Data LCD MF224FDB-T
端末	インテル® Compute Stick STK2m3W64CC
OS	Windows 10 Pro 64bit

4. 在館操作側

在館操作側は、LabVIEW ランタイムをインストールしたタッチパネル端末上で実行させる。機器構成を表 3 に示す。

4-1. 在館操作画面

在館操作画面を図 4 に示す。図中の赤枠は、センター関係者、枠下は VISITOR (共同研究者等) を表示させている。ユーザは、画面のランプ部をタッチすることで、在館の切り替え (ランプの点灯・消灯) を行う。切り替えの際は、判りやすいようにピッと音を鳴らす。



図 4 在館操作画面

4-2. プログラムの内容

在館操作側プログラムのメインとなる処理は、ユーザ操作を受けて Status ファイルの更新を行い、内部サーバにアップロードすることである。プログラムは、まず初期化処理を行い、メイン処理へと進む。メイン処理では 3 つの機能を持つ 4 つのループが並列に動作し、これらループ間のデ

表 4 データ共有用のグローバル変数

名前	型	役割
Members : Member の配列		
Member : ファイル作成の元となる個人データ		
Name	String	名前
Disp. Name	String	画面上での表示名
Status	Boolean	在館かどうか
Time-Record	Boolean	Time-Card を利用するか
In-Time	String	入館日時
Out-Time	String	退館日時
Status : 画面表示に関するデータ		
Disp. Name	String	画面上での表示名
Status	Boolean	在館かどうか

ータ共有に、「Members」「Status」「Time」というグローバル変数を用いる。主なグローバル変数の項目内容を表 4 に示す。

4-2-1. 初期化処理

初期処理は、設定ファイルと在館操作側で保持する Status ファイルのメンバーを比較し、「Members」用データを作成する。「Members」は Status ファイル等を作成するための個人データの配列である。

初期処理は、次のようになっている。

- ① 内部サーバから設定ファイルをダウンロードする。
- ② 設定ファイルを読み込み、在館操作側の Status ファイルとのメンバーの整合性、各個人の状況を確認し、「Members」「Status」にデータを反映させる。
- ③ 「Status」のデータを在館操作画面に表示させる。

4-2-2. メイン処理

メイン処理では、以下の機能を持つループが、それぞれの時間周期やイベントで動作する。

【時刻表示機能ループ】

定期的に現在時刻を取得し、文字列に整形したデータを画面に表示させる。また、文字列時刻を「Time」へ反映させる。

【イベント取得機能ループ】

画面をタッチした時 (イベント発生時) に動作する。画面上の変更値を「Status」の Status へ反映させる。また、タッチ音を鳴らす。

【ファイル処理機能ループ】

2 つのループを並列に動作させ、ファイル処理等を順次処理させるためにキューを用いる。

片方のループでは、定期的に「Status」の Status を確認する。変更があった場合は、「Member」の Status、In-Time や Out-Time に「Time」の値を記録し、Status ファイル更新情報をキューに追加する。また、Out-Time に記録があった場合は、「Member」の Time-Record を確認し、必要なら個人の Time-Card 作成情報をキューに追加する。もう一方のループでは、順次キューからデータ

を取り出し、内容に応じてファイルを作成し、内部サーバへアップロードする処理を行う。

4-3. 掲示メンバー変更

掲示メンバーの変更は、設定ファイルを修正し、プログラム側に再度読み込ませる必要がある。プログラムを再実行し初期処理を行わせることで対応する。

5. 在館者表示側

LAN 接続され LabVIEW ランタイムがインストールされた端末上で、在館者表示側を実行させる。

5-1. 在館者表示画面

在館操作画面と同じレイアウトとし、在館者を視認しやすいようにしている。また、こちらは現在時刻ではなく、在館操作側に変更があった時刻 (Status ファイルの更新時刻) を表示する。

5-2. プログラムの内容

メインとなる処理は、内部サーバ上の Status ファイルをダウンロードし、読み込み、画面上に表示させることである。

定期的に、Status ファイルの更新日時をチェックするループ内で、更新があれば Status ファイルをダウンロードし、画面に表示させる処理を行っている。

6. 短期来訪者の登録機能追加

テスト運用後、短期間来訪者の在館確認にも利用できないか、との要望があがった。数時間~1日程度の来訪者に対して、毎回設定ファイルを修正して対応するのは運用負担が大きい。そのため、VISITOR を共同研究者&長期滞在者と変更し、短期来訪者用ランプ群を追加した(図5の赤枠)。

来訪者ランプには、設定ファイルを介さない登録機能を持たせた。点灯していないランプをタッチすると、登録画面が表示され、名前(必須)、企業名の入力を求める(図6)。名前を登録すると、該当ランプの表示欄に名前が表示される(図7)。また、点灯ランプをタッチすると、登録名は削除される。図7の青枠(点灯ランプ)タッチ後の在館者表示画面を図8に示す。

これらの来訪者機能追加処理は、イベント取得

機能ループに組み入れている。



図5 短期来訪者追加後の在館操作画面

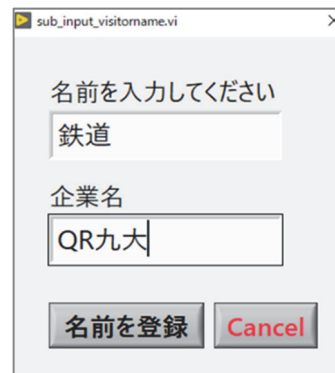


図6 登録画面

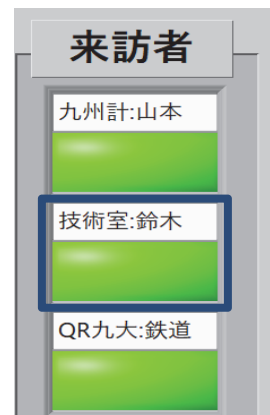


図7 来訪者の名前表示



図8 来訪者点灯ランプタッチ後の在館者表示画面

7. 在館者簡易表示 Web アプリ開発

端末に LabVIEW ランタイムや在館者表示側プログラムをインストールしなくても、実験棟内の様々な場所から手軽に在館者を確認できれば、より便利である。内部サーバ上に Status ファイル (CSV 形式) があるため、それを利用して Web 上で在館者を簡易表示できるようにした。図 9 に Python-Flask で開発したアプリ画面を示す。

内部サーバには Web サービス (Apache) は導入済みであった。今回、Python-Flask を動作させるために、Python3.8 および Python3.8 系の wsgi を導入した。



図 9 Flask による簡易表示画面

8. おわりに

テスト運用を始めるにあたり、実験棟玄関ホールに在館操作側のタッチパネル端末を設置した。以前の在籍盤に似た表示画面のためか、何の説明が無くても掲示メンバーは自然とタッチし、在館操作を始めてくれたことには驚いた。また、在館者表示側も当初はノートパソコン画面を用いていたが、事務室では大型モニタを導入し表示したいなどの声も挙がっている。今回の開発では、様々な意見から、より良いものへと改善する機会を多く持つことができた。大型モニタは近日設置予定である。

現在も順調に運用を続けているが、玄関ホールの在館操作側のタッチパネルを、時折タッチし忘れるとの声もでてきた。設置場所や、玄関ホールにもう一台の在館操作タッチパネル設置し、視界に入る機会を増やすことなども含めて検討していく。

今回、この「在館管理システム」の開発および運用を経て、ここ数年の懸案事項であった在館者確認の一助となることができ、ほっとしている。今後も引き続き関係者の声を聞き、改善に努めていきたい。

謝辞

この「在館管理システム」開発に、様々な意見や要望などを寄せてくださった高温プラズマ理工学研究センター教職員・スタッフの皆さまに、この場を借りて深く感謝いたします。