

Visual Basic for Applications を利用した 放射性試料分析のための前処理法の自動化技術

Automation Technique of Analytical Pre-treatment for Radioactive Sample Measurement
using the Program Written by Visual Basic for Applications

山本 昌彦 家老 克徳 小高 典康 久野 剛彦

Masahiko YAMAMOTO, Yoshinori KARO, Noriyasu KODAKA and Takehiko KUNO

核燃料・バックエンド研究開発部門

核燃料サイクル工学研究所

再処理廃止措置技術開発センター

施設管理部

Facility Management Department

TRP Decommissioning Center

Nuclear Fuel Cycle Engineering Laboratories

Sector of Nuclear Fuel, Decommissioning and Waste Management Technology Development

October 2019

Japan Atomic Energy Agency

日本原子力研究開発機構

JAEA-Technology

本レポートは国立研究開発法人日本原子力研究開発機構が不定期に発行する成果報告書です。
本レポートの入手並びに著作権利用に関するお問い合わせは、下記あてにお問い合わせ下さい。
なお、本レポートの全文は日本原子力研究開発機構ホームページ (<https://www.jaea.go.jp>)
より発信されています。

国立研究開発法人日本原子力研究開発機構 研究連携成果展開部 研究成果管理課
〒319-1195 茨城県那珂郡東海村大字白方 2 番地4
電話 029-282-6387, Fax 029-282-5920, E-mail:ird-support@jaea.go.jp

This report is issued irregularly by Japan Atomic Energy Agency.
Inquiries about availability and/or copyright of this report should be addressed to
Institutional Repository Section,
Intellectual Resources Management and R&D Collaboration Department,
Japan Atomic Energy Agency.
2-4 Shirakata, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195 Japan
Tel +81-29-282-6387, Fax +81-29-282-5920, E-mail:ird-support@jaea.go.jp

© Japan Atomic Energy Agency, 2019

Visual Basic for Applications を利用した
放射性試料分析のための前処理法の自動化技術

日本原子力研究開発機構
核燃料・バックエンド研究開発部門 核燃料サイクル工学研究所
再処理廃止措置技術開発センター 施設管理部

山本 昌彦、家老 克徳*、小高 典康、久野 剛彦

(2019 年 7 月 19 日受理)

グローブボックス、遮蔽セル内で実施している放射性試料分析の前処理法を自動化するため、Microsoft® 社製 Office Excel® Visual Basic for Applications (VBA)でプログラムを作成し、シリンジポンプ、電動ビュレット、フラクションコレクター、電動バルブを制御した。各機器の端子形状と配線に合わせてケーブルを製作し、通信設定に合わせてプログラム中のパラメータを入力した結果、各機器はプログラム通りに操作できることがわかった。そこで、VBA で各機器を逐次的に制御した自動コンディショニング装置と自動分離装置を試作した。自動コンディショニング装置では最大 8 本のカラムへ、硝酸の添加と静置の操作を繰り返し自動で行うことができた。また、自動分離装置では、ストロンチウム(Sr)を吸着・溶離可能な Eichrom Technologies 製 Sr レジンを用いて、模擬高放射性廃液(SHAW: Simulated Highly Active Liquid Waste)中の Sr を分離・回収した。回収液中の Sr 濃度は、作業員の手動操作で分離・回収した試料の測定値、SHAW 中の Sr 濃度調製値と良好に一致した。汎用ソフトである Microsoft® Excel®を利用した本法は、作業の省力化、作業員の被ばく低減、分析の誤操作を防止することができ、自動化に係るコストの大幅な削減も期待できる有効な方法であることがわかった。

本報告書は、日本原子力研究開発機構と長岡技術科学大学において実施された学部学生の実務訓練に基づき行った業務成果に関するものである。

核燃料サイクル工学研究所：〒319-1194 茨城県那珂郡東海村村松 4-33

* 長岡技術科学大学

Automation Technique of Analytical Pre-treatment for Radioactive Sample
Measurement using the Program Written by Visual Basic for Applications

Masahiko YAMAMOTO, Yoshinori KARO*, Noriyasu KODAKA
and Takehiko KUNO

Facility Management Department, TRP Decommissioning Center
Nuclear Fuel Cycle Engineering Laboratories
Sector of Nuclear Fuel, Decommissioning and Waste Management Technology Development
Japan Atomic Energy Agency
Tokai-mura, Naka-gun, Ibaraki-ken

(Received July 19, 2019)

Analytical devices such as syringe pump, electric burette, fraction collector, and electric valve have been controlled by the program written with Microsoft® Excel® Visual Basic for Applications (VBA) to automate the analytical pre-treatment procedure of radioactive sample measurement. It was confirmed that the program written by VBA could control each device by preparing cables and adjusting parameters. Therefore, automatic conditioning and separation systems were made by combining each device and sequentially controlling them with the program. The automatic conditioning system can repeatedly perform conditioning operation like adding nitric acid onto the maximum of 8 columns. The automation separation system can separate and recover strontium (Sr) in simulated highly active liquid waste (SHAW) by using Eichrom Technologies Sr resin. The Sr concentration in recovered sample solution agreed well with the value obtained by manual separation procedure and prepared Sr concentration in SHAW. It is found that the developed automation method, using commercially available Microsoft® Excel®, is effective to reduce labor work, operator's radiation exposure, and to prevent operational error of analysis, together with reducing the cost of automation.

Keywords: Automation, Radioactive Sample, Excel®, Visual Basic for Applications, Analysis

This report contains the work results performed by the undergraduate student practical training between Japan Atomic Energy Agency and Nagaoka University of Technology.

* Nagaoka University of Technology

目次

1. はじめに	1
2. 自動化の対象	2
3. 実験	3
3.1 機器	3
3.2 試薬、及び試料	4
3.3 操作	4
4. 各機器の制御	6
4.1 機器の選定	6
4.2 シリアル通信	6
4.3 シリンジポンプの制御	7
4.4 電動ビュレットの制御	8
4.5 フラクションコレクターの制御	9
4.6 電動バルブの制御	10
5. 自動コンディショニング装置、及び自動分離装置の制作	12
5.1 自動コンディショニング装置によるカラムのコンディショニング操作	12
5.2 自動分離装置による模擬高放射性廃液中のストロンチウムの分離・回収	12
6. 結論	14
謝辞	15
参考文献	16
付録-1 電動バルブの制御プログラム	41
付録-2 自動コンディショニング装置と自動分離装置の制御プログラム	48

CONTENTS

1. Introduction	1
2. Scope of automation	2
3. Experiments	3
3.1 Apparatus	3
3.2 Reagents and sample	4
3.3 Operation	4
4. Control of each analytical device	6
4.1 Selection of analytical device	6
4.2 Serial communication	6
4.3 Control of syringe pump	7
4.4 Control of electric burette	8
4.5 Control of fraction collector	9
4.6 Control of electric valve	10
5. Fabricating automatic conditioning and separation systems	12
5.1 Column conditioning by automatic conditioning system	12
5.2 Separation and recovery of strontium in simulated highly active liquid waste by automatic separation system	12
6. Conclusions	14
Acknowledgments	15
References	16
Appendix-1 Program for electric valve	41
Appendix-2 Program for automatic conditioning and separation systems	48

CONTENTS of Figures

Figure 1	Analytical scheme of plutonium isotopes	17
Figure 2	Operation scheme for resin conditioning	19
Figure 3	Operation scheme for strontium separation	20
Figure 4	Analytical devices selected for each operation	21
Figure 5	Cable connection of syringe pump	23
Figure 6	Syringe pump test program	26
Figure 7	Cable connection of electric burette	27
Figure 8	Electric burette test program	29
Figure 9	Cable connection of fraction collector	30
Figure 10	Fraction collector test program	31
Figure 11	Cable connection of microcomputer and electric valve	32
Figure 12	Schematic and photograph of automatic conditioning system	34
Figure 13	Program flow chart for automatic conditioning system	35
Figure 14	User forms made for automatic conditioning system	36
Figure 15	Schematic and photograph of automatic separation system	37
Figure 16	Program flow chart for automatic separation system	38
Figure 17	User forms made for automatic separation system	39

CONTENTS of Tables

Table 1	Elemental composition of simulated highly active liquid waste	18
Table 2	Pin assignment of serial communication	22
Table 3	Serial communication parameters for each analytical device	24
Table 4	Syringe pump commands for external control	25
Table 5	Electric burette commands for external control	28
Table 6	Valve position by binary control	33
Table 7	Measurement results of strontium	40

This is a blank page.

1. はじめに

東海再処理施設(TRP: Tokai Reprocessing Plant)では、再処理工程で発生したウラン(U)、プルトニウム(Pu)を含む試料、核分裂生成物(FP: Fission Product)を含む高放射性廃液(HAW: Highly Active Liquid Waste)、各施設から発生した低放射性廃液等の様々な放射性試料の分析を行っている。これら放射性試料は、作業者の外部被ばく、内部被ばくを防止するため、グローブボックス、遮蔽セル等の設備内で取り扱っており、分析操作には熟練した作業員を必要とする。TRP は既に使用済燃料の再処理を終了しているが¹⁾、施設の維持管理のため、年間約 6000 件の分析依頼が発生しており、今後予定されている本格的な廃止措置においては、さらに多くの分析依頼が発生する可能性がある。このため、限られた人的資源の中、分析依頼を滞りなく処理するためには、分析に係る各操作の自動化は重要である。

分析操作の自動化は、作業効率の向上、作業員の個人差による分析誤差の減少、ヒューマンエラーの防止等のメリットがあり、様々な分野で試みられている^{2),3)}。また、放射性試料の取り扱いにあたっては、作業者への被ばく低減のメリットもある。TRP でも過去に、産業用の垂直多関節ロボットを用いた U、亜硝酸イオン濃度分析のための吸光度測定の自動化⁴⁾、酸濃度分析のための電位差滴定の自動化⁵⁾、Pu の同位体希釈質量分析のための前処理操作の自動化が行われてきた^{6),7)}。しかし、これらはいずれも機器メーカーが制御プログラムも含めてシステムを製作しており、コストが高く、バグの修正、操作変更時のプログラムをユーザーが編集できないといった課題があった。

本研究では、このような課題を踏まえ、Microsoft® 社製 Excel® Visual Basic for Applications (VBA)*を用いたプログラムによる自動化を検討した。VBA とは、Excel®に含まれる拡張機能を使用するための言語であり、プログラムを作成することで処理の自動化を行うことができる⁸⁾。現在、Excel®は多くのコンピューター(PC: Personal Computer)にインストールされている汎用ソフトであり、VBA は他のプログラミング言語よりも比較的簡単な文法であるため、ユーザーでも容易にプログラムの作成・編集を行うことができる。さらに、分析で汎用されている機器の多くは、外部制御用のシリアル通信端子である RS232C 等が付属されている。そこで、これらの機器を複数個組み合わせて、シリアル通信等で逐次的に制御することで、様々な分析操作の自動化が可能と考えた。これらを踏まえ、VBA と汎用分析機器を組み合わせた放射性試料分析のための前処理方法の自動化を目的として、開発を行った。

※ 「Windows®」、「Excel®」は、米国 Microsoft Corporation 社の、商標または登録商標である。

2. 自動化の対象

TRP の分析対象は、U、Pu、FP であるセシウム-137、ストロンチウム(Sr)-90 等の $\beta\gamma$ 線放出核種、腐食生成物である鉄、クロム、ニッケル等多岐にわたっており、様々な分析法が採用されているが、この中でも表面電離型質量分析法(TIMs: Thermal Ionization Mass Spectrometry)による Pu の同位体分析は、計量管理・保障措置の観点から最も重要な分析の一つである。Figure 1 には、TIMs による Pu の分析前処理スキームを示す。この方法では、試料中に共存する U が同重体として妨害するため、固相抽出剤を用いた分離が行われており、試料分取、希釈、Pu の原子価調整、固相抽出剤のコンディショニング、カラム分離、加熱濃縮・再溶解、質量分析装置用フィラメントへの試料塗布の複数の前処理操作を経て Pu 同位体を測定する。このため、前処理操作を完全に自動化できた場合には、作業量の大幅な低減等の大きな効果が期待できる。

そこで、本研究では TIMs による Pu 同位体分析のための前処理操作を対象とし、さらに当該分析前処理操作の中から、TRP における分析で最も頻度の高い前処理操作である「固相抽出剤のコンディショニング操作」と「カラム分離操作」の自動化を試みた。本報告書では、基礎試験として実施した PC と分析機器間の通信方法の確立、VBA による制御プログラムの作成と分析機器の制御、プロトタイプとして製作した自動コンディショニング装置と自動分離装置を用いたコールド環境における固相抽出剤のコンディショニング操作、模擬高放射性廃液(SHAW: Simulated HAW)中の Sr の分離・回収について述べる。

3. 実験

3.1 機器

(1) PC

装置制御・プログラミング用の PC には、Sony 製 VAIO VGN-G3AAPS (OS: Windows® XP)を使用した。本 PC には RS232C 端子が付属されていなかったため、RS232C 端子付きのケーブルとの接続には、USB シリアル変換ケーブルを使用した。制御プログラムは、Microsoft® 社製 Excel® 2007 の VBA を使用して作成した。標準仕様の VBA では、シリアル通信用の関数が入っていないため、文献に従い EasyComm⁹⁾と呼ばれる VBA 用モジュールを組み込んだのち、プログラムを作成した。なお、Windows® XP は、2014 年に Microsoft® 社からのサポートが終了しているため、PC はネットワークに接続せずに使用した。また、本報告書内で示したプログラムは、Windows® 10、Excel®2013 でも使用可能であることを確認している。

(2) 機器

試料の採取、カラムへの添加には YMC 製 YSP-202 シリンジポンプを使用した。なお、シリンジには、HSW 製 Norm-Ject® 4050.X00V0 プラスチックディスポーザブルシリンジを使用した。また、試薬類の添加には Metrohm 製 665 Dosimat 電動ビュレット(シリンジ容量: 20 mL)、カラムからの各フラクションの回収には Bio-Rad Laboratories 製 Model 2110 フラクションコレクターを使用した。フラクションの回収用試験管には、Bio-Rad Laboratories 製クリアポリスチレンチューブ(外径 13 mm×高さ 100 mm、容量: 9 mL)を使用した。カラムへ添加する試薬類の切り替えには FLOM 製 VA11-108 電動バルブ(12 方)を使用した。電動バルブは、その仕様から PC と直接接続して制御することができなかったため、マイクロコンピュータである Km2Net 製 USB-IO2.0(AKI)を介して使用した。

(3) 接続ケーブル

PC とシリンジポンプの接続には、YMC 製 YSP-PC-001 接続ケーブルを使用した。PC と電動ビュレットの接続用ケーブルは、両端にそれぞれ RS232C 端子(D-sub 9 ピン)、DIN 端子(8 ピン)を設けて自作した。PC とフラクションコレクターの接続用ケーブルは、両端に RS232C 端子(D-sub 9 ピン)を設けて自作した。電動バルブ、マイクロコンピュータ、PC はジャンパーワイヤー(秋月電子通商製 BBJ-65)、USB ケーブルを用いて接続した。

3.2 試薬、及び試料

(1) 試薬

純水は、アドバンテック東洋製 RFD240NA 蒸留水製造装置で蒸留後、MERCK MILLIPORE 製 A10 超純水製造装置で処理したものを使用した。硝酸は、和光純薬工業製硝酸(試薬特級、質量分率:60~61%)を純水で希釈して各硝酸濃度に調整した。Sr 標準溶液は、和光純薬工業製 Sr 標準液(濃度: 1000 ppm)を硝酸で適宜希釈して調製した。

(2) SHAW

SHAW は、東海再処理施設からの高放射性廃液の元素組成を模擬するため、各元素の硝酸塩化合物を溶解し、Table 1 に示す組成となるように調製したものを使用した。

(3) 固相抽出剤

固相抽出剤には、1-octanol 中に調製した 4,4'(5')-di-*t*-butylcyclohexano 18-crown-6(crown ether)をシリカベースの担体表面上に含浸させた Eichrom Technologies 製 Sr レジン(粒径: 100-150 μm)を使用した。これをポリエチレン製ポリプレップ®カラム(Bio-Rad Laboratories)に 1 mL 充填してコンディショニング操作、カラム分離操作に使用した。

3.3 操作

(1) コンディショニング操作

コンディショニング操作は、Figure 2 に示すスキームに従って実施した。Sr レジンを充填したカラムはカラム台にセットしたのち、3 mol/L 硝酸を 8 mL 添加、静置(10 分)の操作を 3 回ずつ繰り返してコンディショニングした。なお、当該操作は、作業員による手動操作と製作した自動コンディショニング装置を用いた自動操作によって行った。

(2) 分離操作

固相抽出剤として使用した Sr レジンは、3 mol/L 硝酸等の高濃度硝酸溶液中で Sr に高い選択性・吸着性を示し、0.01 mol/L 硝酸等の希硝酸溶液で Sr を溶離できる。そこで、SHAW から Sr の分離・回収は、Figure 3 に示すスキームに従って実施した。3 mol/L 硝酸で 1000 倍に希釈した SHAW を 1 mL 採取したのちカラムへ添加し、Sr を固相抽出剤に吸着させた。その後、3 mol/L 硝酸 1 mL を 10 回流してカラムを洗浄し、SHAW 中に共存する Sr 以外の元素を除去した。洗浄後、0.01 mol/L 硝酸 1 mL を 10 回流して Sr をカラムから溶離した。溶離液は、全量をメスフラスコに回収後、1 mol/L 硝酸で 50 mL に定容した。なお、当該操作は、作業員による手動操作と製作した自動分離装置を用いた自動操作によって行った。

(3) Sr の定量

固相抽出剤から回収した試料中の Sr 濃度は、ICP-OES(Perkin Elmer 製 Avio 200)を用いて定量した。検量線は、0.8 mol/L 硝酸ブランク、0.8 mol/L 硝酸で調整した Sr 標準溶液 10 µg/L、50 µg/L、100 µg/L を用いて作成した。Sr の測定波長には、407.771 nm を使用した。ICP-OES は、プラズマガス流量 10 L/min、補助ガス流量 0.2 L/min、ネブライザーガス流量 0.60 L/min、RF 出力 1400 W の条件で使用した。

4. 各機器の制御

本章では、自動コンディショニング装置と自動分離装置を製作するため、コンディショニングとカラム分離操作のスキームを解析して単位操作を求め、当該操作が可能な汎用分析機器を選定した。その後、各機器と PC の配線を調査し、必要となる接続ケーブルを自作した。さらに、通信に必要なパラメータを調整しながら、VBA でプログラムを作成し、PC から各機器の制御を試みた。以下、その結果を示す。

4.1 機器の選定

カラム調整時のコンディショニング操作、SHAW 中の Sr の分離・回収操作を Figure 2、3 に示す。これらを解析すると、基本的な単位操作として、①試料の採取、②試薬類の添加、③フラクションの回収、④添加試薬類の切り替えの作業が挙げられた。これら①～④の操作が可能な機器としては、様々なものがあるが、ここでは比較的安価且つ市販されている汎用分析機器、TRP の分析施設で既に使用されている機器、外部制御用の RS232C 端子等を有し PC により制御可能なものを条件として選定を行った。その結果、Figure 4 に示すように、①試料の採取にはシリンジポンプ、②試薬類の添加には電動ビュレット、③フラクションの回収にはフラクションコレクター、④添加試薬類の切り替えには電動バルブを使用することとした。シリンジポンプは、取り付けしたシリンジによって吸引・吐出し操作を行う機器であり、試料溶液の採取とカラム上への添加を行うことができる。電動ビュレットは、予め専用容器に保管した試薬類をシリンジに吸引後、一定量をカラム上へ吐き出して添加することができる。また、フラクションコレクターでは、カラム出口部からのフラクション溶液を 1 滴ずつ、又は一定量ずつ試験管へ回収することができる。さらに、電動バルブは電動ビュレットからカラムへ添加する試薬類を切り替えることができる。

4.2 シリアル通信

シリンジポンプ、電動ビュレット、フラクションコレクターと PC の通信には、シリアル通信を利用した。電動バルブと PC の通信では、マイクロコンピュータを用いてバイナリ通信を利用した。シリアル通信では、PC と各機器の伝送路上を 1 度に 1 bit ずつ逐次的にデータを送信する。本研究で利用した PC 側 RS232C 端子のピンアサインは、Table 2 に示す通りである。各機器と PC のデータ送受信には、PC 側 RS232C 端子中の 2 番ピン(RxD)、3 番ピン(TxD)を各機器で指定されたピンに接続する必要がある。また、制御する機器によっては、4 番ピン(DTR)、5 番ピン(GND)、6 番ピン(DSR)、7 番ピン(RTS)、8 番ピン(CTS)を用いた信号のやり取りが必要な場合もある。一方、バイナリ通信では、指定されたピンに「ON」、「OFF」となる信号を 2 進数的に送信する。以降の 4.3 節～4.6 節では、PC と各機器の配線、通信設定、制御のために作成した VBA によるプログラムを示す。

4.3 シリンジポンプの制御

(1) PC との接続

本研究で使用したシリンジポンプの背面にはモジュラージャックがあり、これと PC 側 RS232C 端子を YMC 製 YSP-PC-001 接続ケーブルで接続した。なお、ケーブル配線を調査した結果は、Figure 5 に示す通りであった。接続後、シリンジポンプ正面のディスプレイには、外部制御モードを示す逆三角形の記号が表示され、PC と問題なく接続することができた。

(2) 通信設定

シリアル通信では、PC 側 RS232C 端子の COM ポート、通信設定であるデリミタ、ボーレート(通信速度)、パリティ、データビット数、ストップビット数、タイムアウト時間、ハンドシェイクを設定する必要がある。そこで、PC とシリンジポンプの接続にあたっては、各項目を Table 3 に示すように設定し、VBA でプログラム中に記入した。なお、ここに示す各設定値は、PC 側 COM ポート、シリンジポンプ本体でも同じ値に設定する必要があった。このため、PC 側では Windows®のデバイスマネージャーから通信ポートのプロパティを開き、Table 3 と同じ設定値を入力した。また、シリンジポンプ本体では、取扱説明書に従ってパネル操作により設定値を入力した。

本研究で使用したシリンジポンプのボーレートは、300～19200 bps に対応している。ボーレートが速いと通信時間が短くなり、効率的に通信を行うことができる。例えば、ボーレートを最も速い 19200 bps に設定すると、1 bit 当たり約 52 μ s で通信することができる。しかし、このような高速通信条件では、通信時間が短いため、ケーブルの曲げ状態等によって送信データが変化して通信不能になる可能性も考えられた。このため、確実に通信を行うため、ボーレートは設定可能範囲の中間であった 9600 bps に設定した。

(3) 制御プログラム

本研究で使用したシリンジポンプの外部制御用のコマンドを Table 4 に示す。Table 4 のうち、RUN、STP、RESET は、コマンドそのものをシリンジポンプへ送信することで作動するが、DIA、RAT、VOL、DIR はパラメータを必要とするコマンドであり、パラメータとその単位をコマンドと共にシリンジポンプへ送信することで認識される。また、DIR のコマンドは INF と共に送信することでシリンジが順方向である注入操作を行い、WDR と共に送信することでシリンジが逆方向である吸引操作を行う。なお、シリンジポンプへのコマンド送信にあたっては、本体のパネル操作からネットワークアドレスを確認し、当該ネットワークアドレスをコマンドに添付して送信する必要があった。そこで、シリンジポンプのネットワークアドレスを確認した結果、アドレスは「0」であったため、その値をプログラムに記載した。

Figure 6 に直径 10 mm のシリンジを用いて、流速 20 $\mu\text{L}/\text{min}$ 、容量 100 μL を順方向に注入するプログラムを示す(COM ポート番号は 1 と仮定し、その他の設定は Table 3 に示す通り)。シリンジポンプのコマンドは、ASCII コードとして送信し、各コマンドの送信後は 500 ms の待機時間を設け処理を中断させることで、PC から送信したコマンドがシリンジポンプに確実に受信され、動作するように調整した。

作成したプログラムを用いて動作を試みた結果、シリンジポンプはプログラム通りに作動し、VBA で問題なく制御できることがわかった。

4.4 電動ビュレットの制御

(1) PC との接続

本研究で使用した電動ビュレットの背面には、DIN コネクタが取り付けられていた。電動ビュレットの取扱説明書、テスターによる配線確認を行った結果、DIN コネクタは、Figure 7 に示す配線で PC と接続できることがわかった。そこで、これらの接続が可能となるように配線し、両端に RS232C 端子と DIN 端子を設けたケーブルを作製した。なお、電動ビュレットの制御にあたっては、通信状態を確実に担保するため、PC の電源投入後に電動ビュレットの電源を投入する必要がある。

(2) 通信設定

電動ビュレットのシリアル通信設定は、取扱説明書に従い Table 3 に示す値に調整した。

(3) 制御プログラム

使用した電動ビュレットの外部制御用コマンドを Table 5 に示す。電動ビュレットもシリンジポンプと同様に、コマンドそのものを送信することで作動するものと、パラメータを必要とするコマンドに分けられる。Table 5 であれば、REM ON、REM OFF、G、S、F、DIR、DOS は前者であり、その他の VUP、VDW、VLI は後者で、送信にはパラメータである数値の設定が必要となる。

Figure 8 に電動ビュレットの動作プログラムの例を示す。このプログラムは、電動ビュレットを外部制御モード、DOS モードに移行させたのち、試薬類をシリンジへ吸引したのち、吐出させる操作を行うものである。吸引時の流速は 30 mL/min 、吐出時の流速は 40 mL/min 、吐出容量は 25 mL に設定した。本研究で使用した電動ビュレットで設定可能な流速範囲は、シリンジの容量(1、5、10、20、50 mL)によって異なるため、設定時はシリンジ容量を確認しておく必要がある。なお、取り付けしたシリンジの容量は、電動ビュレットにて自動認識されるため、交換の際に設定するパラメータ等はない。また、電動ビュレットのコマンド「REM ON」はコマンドを送信後、一定時間経過しないと本体へ反映されなかったため、「REM ON」コマンド送信後に 4000 ms の待機時間を設けた。その他のコマンドについても動作安定化のため、送信後に待機時間 500 ms を設けるとともに、吸引操作のコ

マンドである「F」送信後は、60000 ms の待機時間を設けて本体の吸引操作が完全に終了後、次の操作に移行できるように調整した。

これらの結果、電動ビュレットはプログラム通りに作動し、VBA で問題なく制御できることがわかった。

4.5 フラクションコレクターの制御

(1) PC との接続

本研究で使用したフラクションコレクターの背面には、RS232C 端子が取り付けられていた。フラクションコレクターの取扱説明書、テスターによる配線確認を行った結果、Figure 9 に示す配線で PC と接続できることがわかった。そこで、これらの接続が可能となるように両端に RS232C を設けたケーブルを作製した。このケーブルを用いて、フラクションコレクターと PC を接続すると外部制御モードに移行し、フラクションコレクター本体パネルの操作ができなくなった。

(2) 通信設定

フラクションコレクターの動作にあたっては、PC から 100 ms 以上のパルス信号を受信する必要がある。このため、パルス信号を確実に送受信できるように、シリアル通信設定のボーレートはシリッジポンプ、電動ビュレットよりも遅い 1200 bps に設定した。その他のパラメータは、シリアル通信で一般的に使用される値として、Table 3 に示す通りに設定した。

(3) 制御プログラム

フラクションコレクターの操作には、前述したように PC から 100 ms 以上のパルス信号を送信する必要がある。一方、VBA によるシリアル通信では、パルス幅を設定してデータを送信することはできない。そこで、ASCII コードによるコマンド送信時のデータ量を調整することで、100 ms 以上の通信時間を確保して、フラクションコレクターの操作を試みた。

半角英数字 1 文字当たりのデータ量は 1 Byte であり、これは 8 bit に相当する。フラクションコレクターのボーレートは 1200 bps であるため、半角英数字 15 文字(データ量: 120 bit)を送信すれば、通信時間は理論上 100 ms となる。これを考慮して作成したフラクションコレクターの動作プログラムを Figure 10 に示す。なお、プログラム中で送信する半角英数字は「o」とし、これを 47 個連続して送信することとした。これは「o」が ASCII コードの 16 進数で「6f」に相当し、2 進数では「01101111」となる。2 進数表示の「0」を信号 OFF、「1」を信号 ON とすると、「01101111」は ON の状態が比較的長く続き、疑似パルスとしてフラクションコレクターに認識させることができるのではないかと考えた。また、

PC とフラクシオンコレクターの通信、動作を安定化させるため、「o」を 47 個連続して送信することで通信時間を約 300 ms に調整した。

これらの結果、フラクシオンコレクターはプログラム通りに作動し、VBA で問題なく制御できることがわかった。

4.6 電動バルブの制御

(1) PC との接続

本研究で使用した電動バルブは、複数の出力(ON)又は非出力(OFF)の信号を受信することで動作するものであった。このため、電動バルブの操作には PC から 5 個の ON/OFF 信号を同時に送信する必要があった。一方、VBA では、同時に複数個の信号送信をプログラムすることはできない。そこで、マイクロコンピュータを Figure 11 に示すように、PC と電動バルブ間に設置し、PC を操作してマイクロコンピュータから複数個の ON/OFF 信号を送信することで電動バルブの制御を試みた。

マイクロコンピュータは、USB インターフェースを搭載した入出力基盤であり、PC からコマンドを送信することで複数個の ON/OFF 信号を同時に出力することができる。Figure 11 に示すように、マイクロコンピュータの J1 側には J1-0～J1-7 の 8 個、J2 側には J2-0～J2-3 の 4 個の入出力ピンが設けられている。各入出力ピンは、ON の時に 5 V、OFF の時に 0 V が印可されるように調整した。電動バルブとは Figure 11 に示す配線で、マイクロコンピュータ J1 側の J1-0～4、J1-7 の入出力ピンを電動バルブ背面の RS232C 端子の 8 番、21 番、9 番、22 番、7 番、19 番ピンにそれぞれ接続した。また、電動バルブとの間には LED ライトを設け、プログラム実行時にどのピンが ON、OFF になっているか目視確認ができるようにした。なお、マイクロコンピュータと PC は、USB ケーブルで接続した。

(2) 通信設定

電動バルブはマイクロコンピュータを介してバイナリ通信で制御したため、シリアル通信用の通信設定は必要なかった。

(3) 制御プログラム

① 電動バルブのバルブ位置の変更方法

本研究で使用した電動バルブは 12 方バルブであり、バルブ位置は No.1～12 に変更することができる。バルブ位置はバイナリ入力により制御され、電動バルブ本体背面の RS232C 端子は、初期状態では全て電圧が OFF の状態になっており、マイクロコンピュータ等からの信号で全ての端子を ON の状態に変更する必要がある。その後、バルブ位置を変更する場合は、Table 6 に示す端子を OFF に調整することで、所定のバルブ位置に変更することができる。一例として、バルブ位置を No.3 に変更する場合は、電動バルブの RS232C の

8 番、21 番、7 番ピンを OFF にする必要がある。なお、電動バルブ側 RS232C 端子の各ピンの役割は以下に示す通りであった。

- 8、21、9、22 番ピン ・ バイナリ入力により、バルブ位置のポジションを指定
- 7 番ピン ・ ・ ・ ・ ・ バイナリ入力により、バルブ位置を指定ポジションに変更
- 19 番ピン ・ ・ ・ ・ ・ 電圧を印加させることで、外部制御モードへ移行

② マイクロコンピュータによる電動バルブの制御

電動バルブに接続したマイクロコンピュータの J1 側入出力端子へのバイナリ入力では「1」が ON、「0」が OFF に相当する。J1 側の端子数は 8 個なので、最大データ量は 8 bit となる。上述に示した例のように、電動バルブのバルブ位置を No.3 に変更するためには RS232C 端子 8 番ピン、21 番ピン、7 番ピンに接続した J1-0、J1-1、J1-4 を OFF、それ以外を ON の状態に調整する必要がある。これをバイナリで表すと「11101100」となり (Table 6)、マイクロコンピュータからこの値を電動バルブに送信することで、バルブ位置を No.3 に変更できる。

③ プログラム

電動バルブは、USB 接続によるマイクロコンピュータを介して制御する必要があった。しかし、VBA の標準仕様では USB 接続機器にデータを送受信することができない。このため、プログラム中に宣言文を記入し^{10)・12)}、マイクロコンピュータのベンダー ID とプロダクト ID を入力することで USB 接続による通信を行った。

また、PC からマイクロコンピュータへのデータ送信は 16 進数法で行う必要があった。例えば、電動バルブのバルブ位置を No.3 に変更する場合は、バイナリ「11101100」に相当する 10 進数である「236」を入力し、VBA 内で Hex 関数を用いて 16 進数である「EC」に変換してマイクロコンピュータに送信した。さらに、プログラム中に「Amount」と呼ばれる可変長引数を設け、8 bit で表現できる最大数 255 から J1-0～7 のピン番号に相当する 2 の乗数をそれぞれ引くことで、J1 側ピン番号を指定して、バイナリ入力を容易にできるように調整した。これらを考慮しながら作成した電動バルブの動作プログラムを付録-1「電動バルブの制御プログラム」に示す。

当該プログラムを用いて PC から電動バルブを操作し、バルブ位置を No.3 へ変更した結果、電動バルブは問題なく作動されることが確認できた。これらの結果より、VBA で作成したプログラムで、電動バルブを制御できることがわかった。

5. 自動コンディショニング装置、及び自動分離装置の制作

本章では、前章で VBA を用いて制御した各機器を組み合わせ、カラムのコンディショニング操作を自動化した自動コンディショニング装置、及びカラムによる分離操作を自動化した自動分離装置を製作した。さらに、自動分離装置を用いて、SHAW 中の Sr の分離操作を試みた。以下、これらの結果を示す。

5.1 自動コンディショニング装置によるカラムのコンディショニング操作

Figure 2 を確認すると、カラムのコンディショニング操作では、3 mol/L 硝酸をカラムに添加後、カラムから硝酸が流れ出るまで 10 分間静置する操作を 3 回繰り返す必要がある。また、通常の分析においては、測定結果のバラつきを確認する目的で複数個のカラム操作を行うことが多く、コンディショニング操作も複数個のカラムについて行う必要がある。そこで、電動ビュレットと電動バルブを組み合わせ、最大 8 本のカラムがコンディショニング可能な自動コンディショニング装置を製作した。

製作した自動コンディショニング装置の概要を Figure 12、自動コンディショニング装置用動作プログラムのフローチャートを Figure 13 に示す。なお、プログラムの詳細は、付録-2「自動コンディショニング装置と自動分離装置の制御プログラム」に示す通りである。

自動コンディショニング装置では、コンディショニングに使用する試薬瓶中の 3 mol/L 硝酸を電動ビュレットで吸引して吐出し、電動バルブで所定のカラムに振り分けるように調整した。また、VBA の機能の一つであるユーザーフォームを設けることで、3 mol/L 硝酸のカラムへの吐出流速、カラムに充填する硝酸の容量、コンディショニングを行うカラムの本数、コンディショニングの回数、通液後の静置時間の入力画面を作成し、ユーザーがパラメータとして入力後、自動で Excel®のワークシート中に記載されるようにした(Figure 14)。さらに、プログラムは機器毎にモジュールを分けて作成することで、プログラム編集時等に使いやすいようにした。

これらの結果、製作した自動コンディショニング装置は、プログラム通りに作動し、Sr レジンを充填した 8 本のカラムのコンディショニング操作を問題なく自動で行うことができた。

5.2 自動分離装置による模擬高放射性廃液中のストロンチウムの分離・回収

Figure 3 を確認すると、Sr レジンをを用いて SHAW 中の Sr をマトリックスから分離・回収するためには、試料溶液を採取、カラムに添加後、3 mol/L 硝酸で洗浄したのち、0.01 mol/L 硝酸を添加して Sr を回収する必要がある。そこで、シリンジポンプ、電動ビュレット(2 台)、フラクションコレクター、電動バルブを組み合わせた自動分離装置を製作した。

製作した自動分離装置の概要を Figure 15、自動分離装置用動作プログラムのフローチャートを Figure 16 に示す。なお、プログラムの詳細は、付録-2「自動コンディショニング装置と自動分離装置の制御プログラム」に示す通りである。

自動分離装置では、試料採取後のシリンジを 3 mol/L 硝酸、純水で洗浄できるように調整した。また、自動コンディショニング装置と同様にユーザーフォームを作成して入力画面を設け、ユーザーが試料採取時、3 mol/L 硝酸添加時、0.01 mol/L 硝酸添加時の流速、容量、通液後の待機時間、シリンジ洗浄時の硝酸、純水の流速と容量を入力できるようにした (Figure 17)。なお、プログラム作動中は、カラム洗浄、溶離、シリンジ洗浄のどの操作を行っているかわかるように、操作内容をユーザーフォームで表示できるようにした。

その結果、製作した自動分離装置は正常に動作し、SHAW 中の Sr を自動で分離・回収することができた。Table 7 に SHAW 中の Sr を自動分離装置で分離・回収した試料、作業員の手動操作により分離・回収した試料をそれぞれ ICP-OES で測定した結果と SHAW 中の Sr 濃度調製値を示す。自動分離装置で分離・回収した試料中の Sr 濃度は 1.0 g/L であり、作業員が手動操作で分離・回収した試料中の Sr 濃度である 1.2 g/L、SHAW の調製値である 1.1 g/L と良好な一致を示した。また、自動分離装置及び作業員の手動操作で分離・回収した試料中には、SHAW に大量に含まれる Na のピークは確認されなかった。これらの結果より、製作した自動分離装置を用いることで、手動分離と同等の性能で SHAW 中の Sr を分離・回収できることがわかった。

自動分離装置を使用した場合と作業員の手動操作では、両者とも Sr の分離・回収に約 40 分の時間を要した。放射性試料の取り扱いは、内部被ばくや外部被ばくの防止のため、グローブボックスや遮蔽セル内で行う必要があり、通常 2 人 1 組で作業を実施している。このため、作業員が手動操作でカラム分離を行った場合、その作業量は約 80 人・分となる。一方、自動分離装置では、グローブボックス又は遮蔽セル内に装置をセットアップすれば、ボタン操作のみとなるため、1 人でも作業を行うことができる。このため、作業量は約 40 人・分となり、作業員の手動操作と比べて作業量は半分と大幅に低減でき、作業の効率化を図れることがわかった。また、自動分離装置ではボタン操作のみでカラム分離を行うことができるため、作業員のレベルによる分析誤差が発生せず、プログラムで定められた動作のみを行うためヒューマンエラーの発生もない。さらに、作業員によるグローブ操作等が不要となるため、試料が高線量の場合は被ばくの低減化を図ることができる。

6. 結論

本研究では、グローブボックス、遮蔽セル内で実施している放射性試料分析に係る前処理操作を自動化するための基礎試験として、VBA で作成したプログラムを用いて、シリンジポンプ、電動ビュレット、フラクションコレクター、電動バルブの各機器の制御を試みた。各機器の通信設定に合わせてケーブル、プログラム中の通信パラメータ等を調整することで、VBA でもこれらの機器を問題なく操作できることがわかった。そこで、各機器を組み合わせ、プログラムで逐次的に制御することで自動コンディショニング装置と自動分離装置を試作した。各装置はプログラム通りに作動し、自動コンディショニング装置では、Sr レジンを充填したカラム上に洗浄用硝酸の添加、静置の繰り返し操作を自動で行うことができ、最大 8 本のカラムを同時にコンディショニングすることができた。また、自動分離装置では、SHAW の Sr レジン充填カラムへの添加、洗浄用硝酸の添加、カラムから Sr の分離操作を自動で行うことができた。回収液中の Sr 濃度は、手動により分離・回収したもの、SHAW の調製値と有意な差はなく、製作した自動分離装置で SHAW 中の Sr を問題なく分離・回収できることがわかった。

これらの結果から、汎用ソフトである VBA でプログラムを作成することで市販の分析機器を制御可能であり、これらを組み合わせることで自動コンディショニング装置と自動分離装置を製作することができ、VBA による分析機器の制御が放射性試料分析のための前処理法の自動化に有効なものであることがわかった。

本研究では、TIMS による Pu の同位体分析のための前処理法を自動化することを目的としている。これまで得られた試験結果より、全ての前処理操作を自動化するためには、以下の課題があることがわかった。

① 制御プログラムの最適化

プログラム製作では、様々な条件で所定の操作を繰り返し実施し、バグを発見して適宜修正する必要がある。本研究では、自動コンディショニング装置と自動分離装置を試作し、Sr を用いた分離試験を実施したが、試行回数が少なく未発見のバグがプログラム中に存在している可能性がある。このため、今後、試行回数を増加させてバグを発見し、適宜修正を行う必要がある。

② グローブボックス内設置のための機器の調整

Pu 等の放射性試料は、被ばく防止の観点からグローブボックス等内で取り扱う必要がある。このため、自動化に係る分析機器もグローブボックス内に設置しなければならない。この際、PC と各機器はグローブボックス内外で通信するため、本研究で使用した接続ケーブルよりも長いものを使用する必要がある。ケーブル線を延長した場合、電氣的ノイズが発生して通信状態に影響を与える可能性があるため、通信速度等のパラメータを調整して通信状態を安定させる必要がある。

また、本研究では、カラム上に試料溶液、洗浄液、溶離液を添加して分離操作を行ったが、カラム内の樹脂中に空隙が存在していた場合、添加溶液が通液されず目詰まりが起こる。手動操作では、詰まりの発生時に作業員が樹脂充填状態を調整することが可能だが、自動操作では詰まりが発生してもプログラムが進行し、添加溶液がカラムから溢れ出てしまう可能性が考えられた。このため、樹脂中に空隙が存在していても通液が可能となるように、気密を確保したオンラインカラム等の適用を検討する必要がある。

③ コンディショニング、カラム分離以外の操作の自動化

TIMS による Pu の同位体分析の前処理操作を完全に自動化するためには、コンディショニング、カラム分離以外に、試料の希釈、原子価調整、加熱濃縮、溶解、試料のフィラメントへの塗布の操作を自動化する必要がある。試料の希釈、溶解は、本研究でも使用した電動ビュレットを用いることで自動化が可能と考えられる。一方、原子価調整、加熱濃縮は、電動ビュレットに加えて、ヒーター、温度調節器等を制御する必要があり、これらの制御プログラムも作成しなければならない。また、試料のフィラメントへの塗布、さらに各操作終了後の試料の運搬にあたっては、ロボットアーム等の使用も検討する必要がある。

Excel®は多くの PC にインストールされているソフトウェアであり、VBA は他のプログラム言語よりも比較的簡単な構文でプログラムを作成することができる。このため、VBA を利用した自動化装置では、ユーザーでプログラムの編集等を容易に行うことができ、従来の自動化システムよりも制作コストを大幅に削減できると考えられる。今後、上記課題を検討して TIMS による Pu の同位体分析に係る前処理操作を自動化すると共に、他の分析法へも適用していくことが期待される。

謝辞

本研究は、日本原子力研究開発機構と長岡技術科学大学において実施された学部学生の実務訓練に基づき行ったものです。再処理廃止措置技術開発センターの栗田 勉 施設管理部長、長岡技術科学大学の伊藤 治彦 准教授には、実務訓練生の受け入れにあたり多大なご協力を頂きましたことを深く感謝いたします。また、本研究の実施にあたっては、分析課の稲田 聡 マネージャー、渡邊 伸久 技術副主幹、堀籠 和志 主査、下越 三輝 氏、秋山 泰之 氏から活発な議論、励まし、助言を頂きましたことを深く感謝いたします。

参考文献

- 1) 岡野正紀, 秋山和樹, 田口克也, 永里良彦, 大森栄一, 東海再処理施設の廃止措置計画の概要, デコミッションング技報, 57, 2018, pp.53-64.
- 2) 三井泰裕, 研究所における分析自動化システム, 日立評論, 59, 1977, pp.59-64.
- 3) 日本機械工業連合会, 日本分析機器工業会, 平成 19 年度次世代産業の実用化を加速する分析の自動化に関する調査報告書, 日機連 19 高度化-5, 2008, 156p.
- 4) 鈴木弘之, 山田敬二, 田村三郎, 池田久, 久野祐輔, 多成分自動分析システムの開発(I) 吸光光度法自動分析装置の開発, 動燃技報, 94, 1995, pp.91-94.
- 5) 稲田聡, 鈴木弘之, 鴨志田政之, 神長一博, 庄司和弘, 池田久, 久野祐輔, 多成分自動分析システムの開発(II) 電位差滴定法自動分析装置の開発, PNC-TN8410 96-266, 1996, 67p.
- 6) 佐藤宗一, 西田恭輔, 阿部勝男, 久野祐輔, 秋山孝夫, JASPAS JC-7; Automated gravimetric sample preparation system (AGSS) for safeguards analysis, PNC-TN8410 89-062, 1990, 145p.
- 7) N.Surugaya, T.Hiyama, M.Watahiki, Automated gravimetric sample pretreatment using an industrial robot for the high-precision determination of plutonium by isotope dilution mass spectrometry, Anal. Sci., 24, 2008, pp.739-744.
- 8) 寺坂宏一, 化学系学生のための Excel/VBA 入門 -Office 2007 対応-, コロナ社, 2009, 181p.
- 9) 櫻木嘉典, Excel を用いた計測制御入門(改訂新版), 電気書院, 2015, pp.186-190.
- 10) K.ISSEI, dinop.com, PSoC を Excel のマクロ機能(VBA)だけで USB 制御する, https://www.dinop.com/elec/psoc_usb_txrx_by_excel.html (参照 2019 年 5 月 22 日).
- 11) Keysight Technologies, 初心者向け Excel/VBA による測定器制御プログラム 超入門編, <http://literature.cdn.keysight.com/litweb/pdf/5992-0691JAJP.pdf> (参照 2019 年 5 月 22 日).
- 12) 桑野雅彦, パソコンによる汎用 I/O 制御を可能にする USB 汎用インターフェースキット, CQ 出版, 2004, pp.19-34.

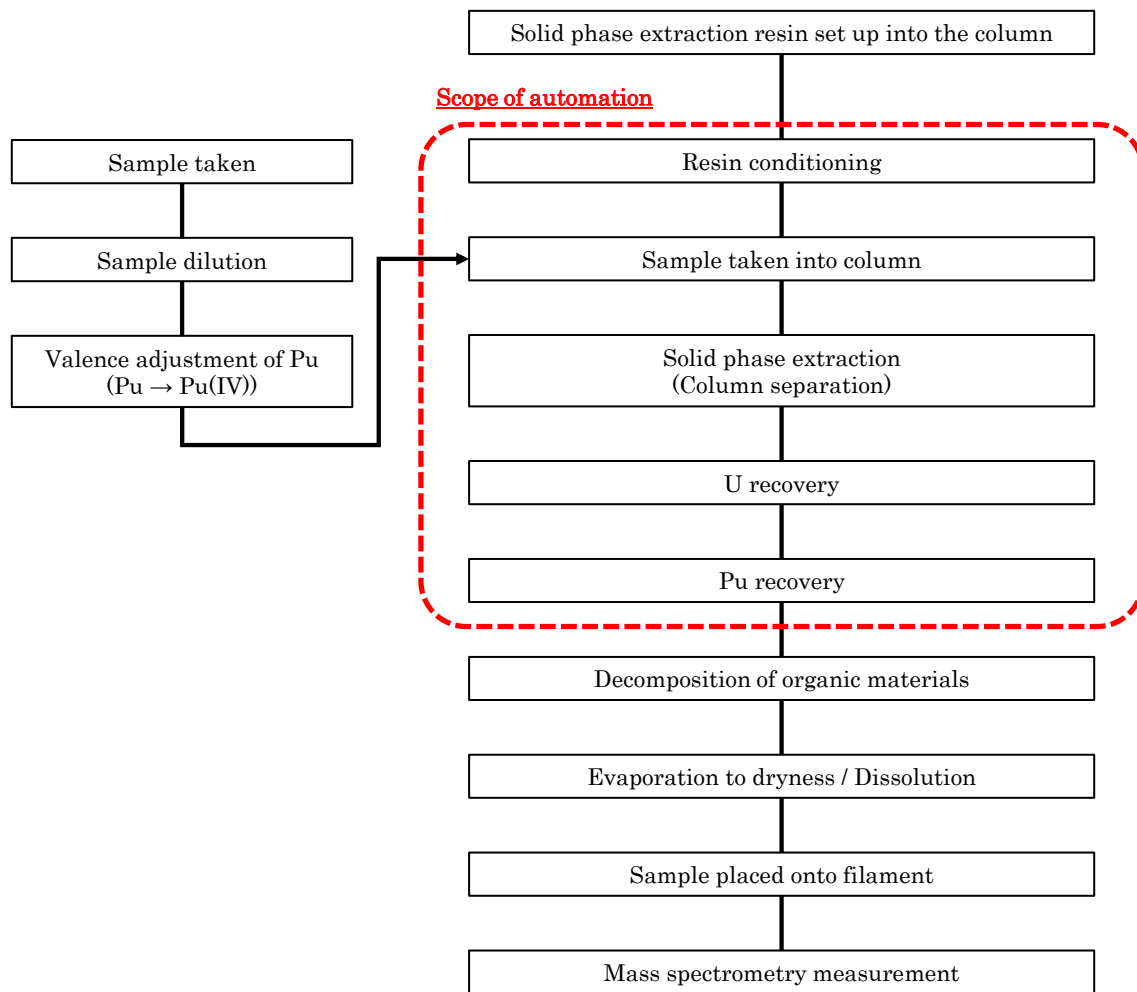


Figure 1 Analytical scheme of plutonium isotopes

The measurement is performed by thermal ionization mass spectrometry. The dashed line indicated the scope of automation examined in this study.

Table 1 Elemental composition of simulated highly active liquid waste

The solution was prepared by dissolving nitrate of each element. Acid concentration was adjusted to be 1.38 mol/L.

Element	Concentration (g/L)	Element	Concentration (g/L)
Na	23	Rh	0.52
P	0.13	Pd	1.7
K	0.094	Ag	0.10
Cr	0.33	Zn	0.029
Fe	1.7	Te	0.64
Ni	0.64	Cs	4.8
Co	0.12	Ba	2.2
Sr	1.1	La	3.1
Y	0.73	Ce	5.1
Zr	4.7	Pr	2.2
Mo	3.1	Nd	8.9
Mn	1.0	Sm	1.4
Ru	2.2	Gd	5.7

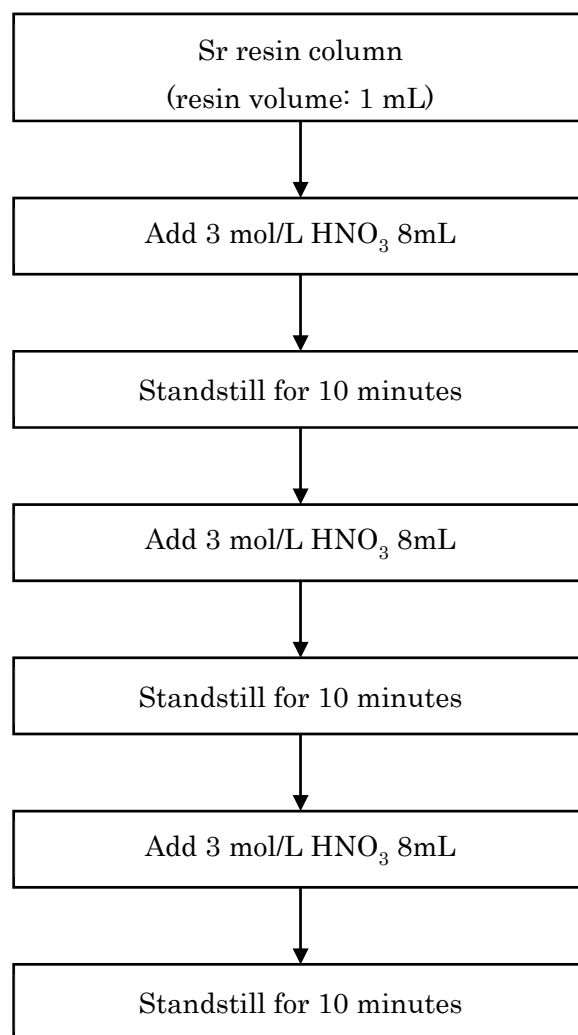


Figure 2 Operation scheme for resin conditioning

Both manual operation by operator and automatic operation by automatic conditioning system were performed according to the conditioning scheme.

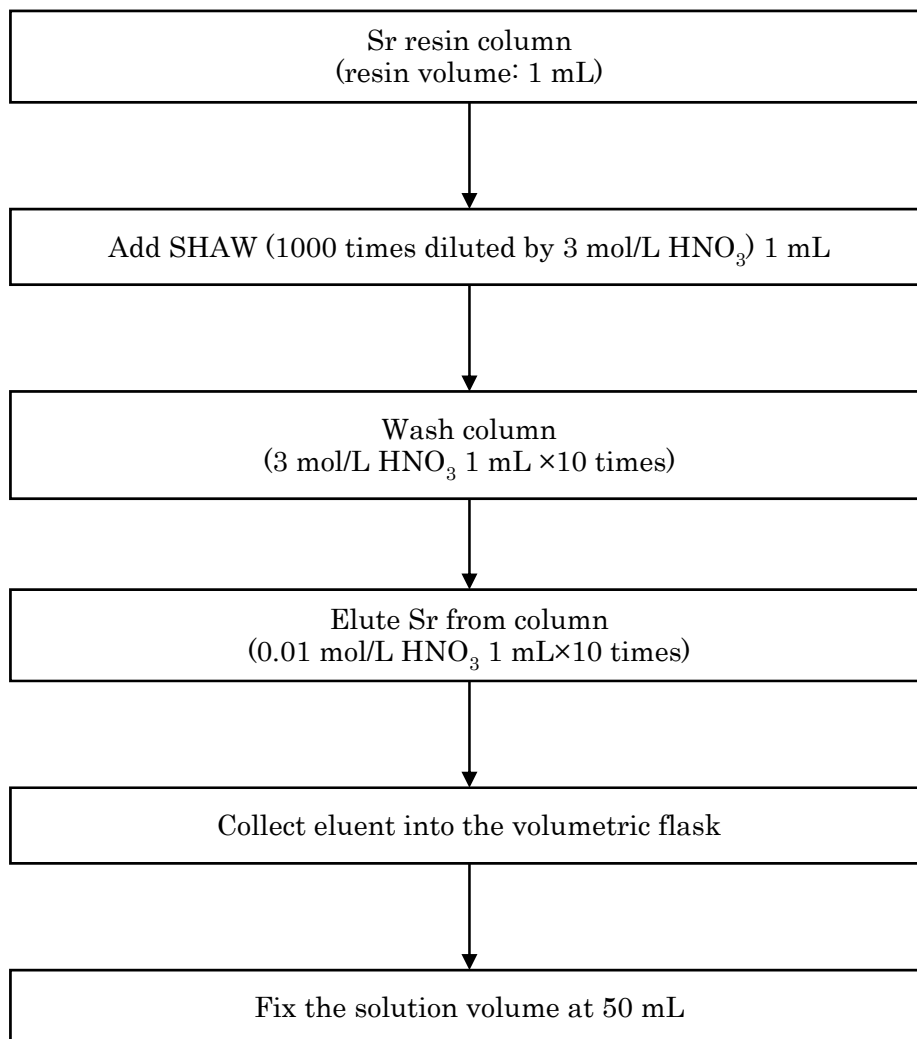


Figure 3 Operation scheme for strontium separation

Both manual operation by operator and automatic operation by automatic separation system were performed according to the separation scheme.

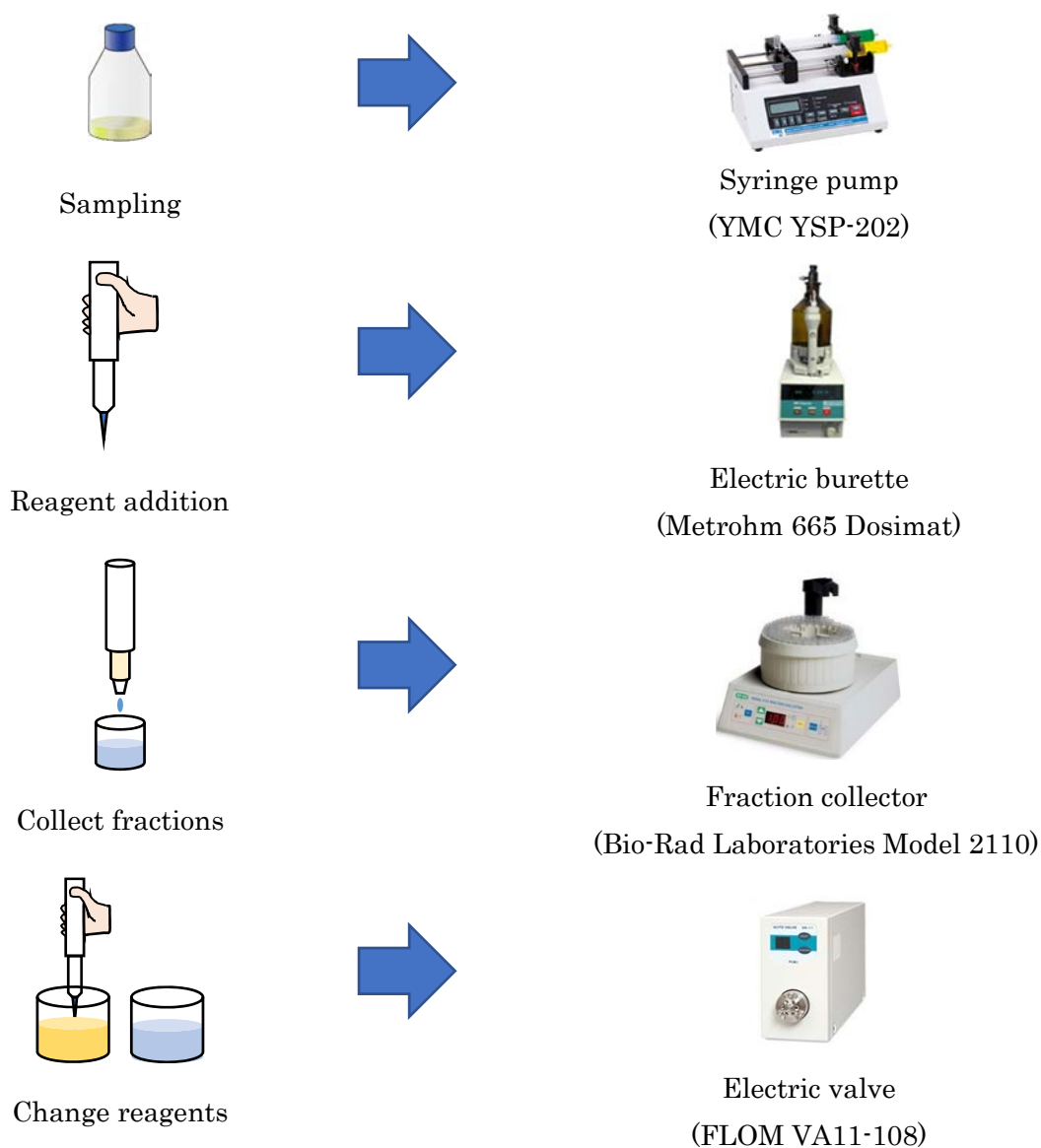


Figure 4 Analytical devices selected for each operation

All devices were already used in analytical laboratory of TRP and had communication connector for external control.

Table 2 Pin assignment of serial communication

Cable colors in the table were used throughout all experiments in this study.

Pin	Function	Signal	Signal Type	Color
1	Carrier Detector	DCD	IN	Black
2	Receive Data	RxD	IN	Brown
3	Transmit Data	TxD	OUT	Red
4	Data Terminal Ready	DTR	OUT	Orange
5	Signal Ground	GND	---	Yellow
6	Data Set Ready	DSR	IN	Green
7	Request To Send	RTS	OUT	Blue
8	Clear To Send	CTS	IN	Purple
9	Ring Indicator	RI	IN	Gray/White

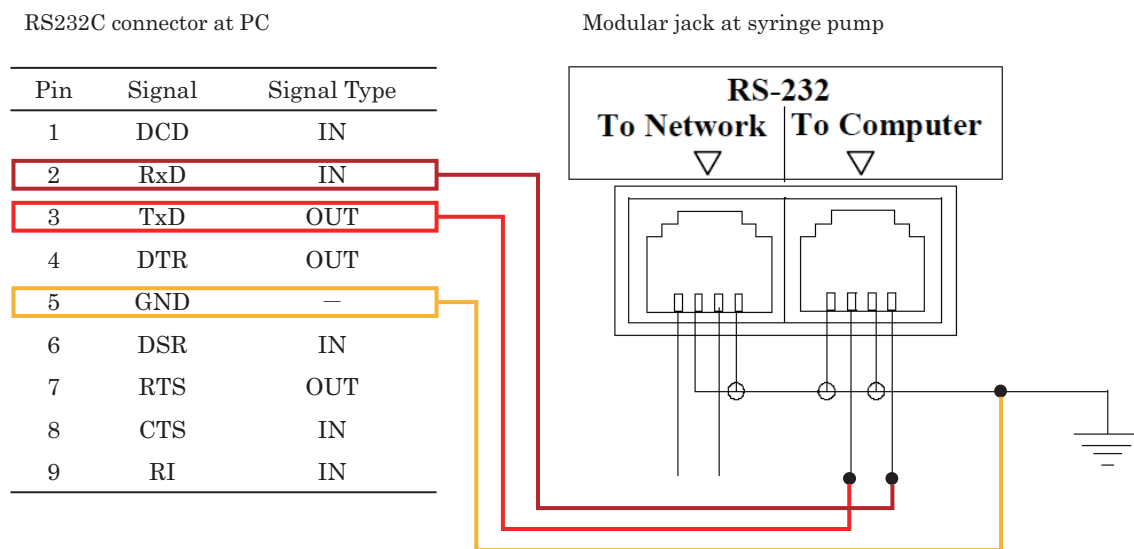


Figure 5 Cable connection of syringe pump

The RS232C connector was used for PC and the modular jack connector was used for syringe pump.

Table 3 Serial communication parameters for each analytical device

Devices are syringe pump, electric burette, and fraction collector. Set values were written in VBA program for external control.

Parameter	Syringe pump	Electric burette	Fraction collector
Delimiter	Cr	CrLf	CrLf
Baud rate	9600 bps	9600 bps	1200 bps
Parity	None	Even	None
Data bit	8 bit	7 bit	8 bit
Stop bit	1 bit	1 bit	1 bit
Time out	1500 ms	1500 ms	1500 ms
Handshake	None	XOn/XOff	None

Table 4 Syringe pump commands for external control

The command was sent from PC to syringe pump by ASCII transmission.

Command	Function
RUN	Pump start
STP	Pump stop
DIA	Set diameter of syringe
RAT	Set flow rate
VOL	Set volume
DIR(WDR/INF)	Set direction (withdraw or forward)
RESET	Pump reset

```

Sub test1() 'シリンジポンプのプログラム

ec.COMn = 1 'ポート番号の設定(ポート番号を1に設定)
ec.Delimiter = "Cr", 'リミタの設定(デリミタを0に設定)
ec.Setting = "9600,n,8,1", '通信条件の設定(ボーレート9600bps、パリティなし、データビット長8bit、ストップビット長1bitに設定)
ec.AsclLineTimeOut = 1500 'タイムアウトの設定(タイムアウトを1.5秒に設定)
ec.HandShaking = "N", 'ハンドシェイクの設定(ハンドシェイクなしに設定)
ec.WAITms = 5000 '処理の中断時間を設定(コマンド送信後、5秒間中断に設定)

ec.AsclLine = "RESET" 'シリンジポンプ本体パラメータのリセット
ec.WAITms = 500 '処理の中断時間を設定(コマンド送信後、0.5秒間中断に設定)

ec.AsclLine = "0 DIA 10" 'シリンジの直径を設定(ネットワークアドレスは0、シリンジの直径を10mmに設定)
ec.WAITms = 500 '処理の中断時間を設定(コマンド送信後、0.5秒間中断に設定)

ec.AsclLine = "0 RAT 20 UM" '流速と単位を設定(ネットワークアドレスは0、流速は20、単位はμL/minに設定)
ec.WAITms = 500 '処理の中断時間を設定(コマンド送信後、0.5秒間中断に設定)

ec.AsclLine = "0 VOL UL" '注入容量の単位を設定(ネットワークアドレスは0、注入容量の単位はμLに設定)
ec.WAITms = 500 '処理の中断時間を設定(コマンド送信後、0.5秒間中断に設定)

ec.AsclLine = "0 VOL 100" '注入容量の単位を設定(ネットワークアドレスは0、注入容量を100 μLに設定)
ec.WAITms = 500 '処理の中断時間を設定(コマンド送信後、0.5秒間中断に設定)

ec.AsclLine = "0 DIR INF" 'シリンジの動作方向を設定(ネットワークアドレスは0、シリンジの動作方向を注入に設定)
ec.WAITms = 500 '処理の中断時間を設定(コマンド送信後、0.5秒間中断に設定)

ec.AsclLine = "0 RUN" 'ポンプ動作開始

End Sub

```

Figure 6 Syringe pump test program

The communication setting was set according to Table 3. Network address was set as 0.

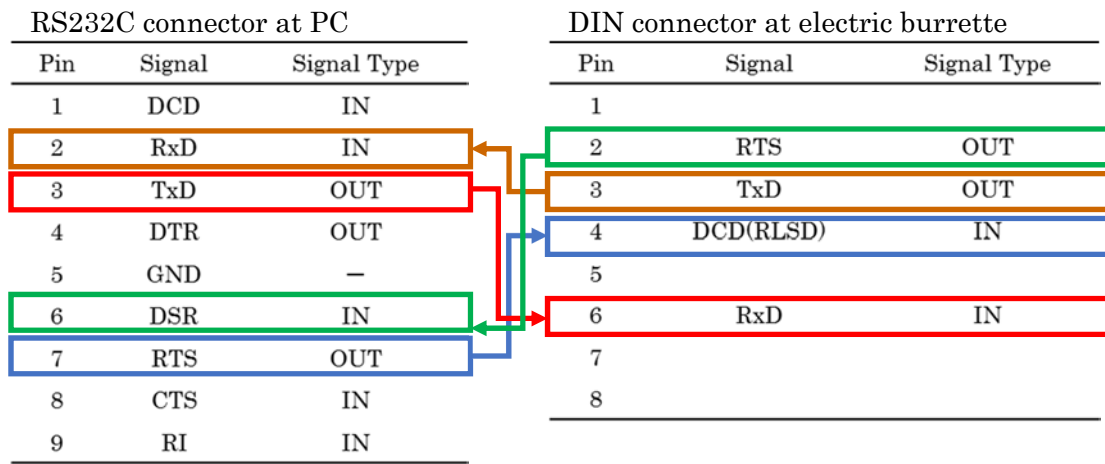


Figure 7 Cable connection of electric buret

The RS232C connector was used for PC and the DIN connector was used for electric burette.

Table 5 Electric burette commands for external control

The command was sent from PC to electric burette by ASCII transmission.

Command	Function
REM ON	Remote control on
REM OFF	Remote control off
G	Go (Inject reagents)
S	Stop (Stop injection/refill)
F	Fill (Refill reagents)
VUP	Rate up
VDW	Rate down
VLI	Pipetting volume
DIR	Repetitive dispensing mode
DOS	Dosing mode

```

Sub test2() '電動ビュレット(Dosimat)のプログラム

ec.COMn = "1" 'ポート番号の設定(ポート番号を1に設定)
ec.Delimiter = "CrLf" 'デリミタの設定(デリミタをCrLfに設定)
ec.Setting = "9600,e,7,1" '通信条件の設定(ボーレート9600bps、パリティEven、データビット長7bit、ストップビット長1bitに設定)
ec.WAITms = 500 '処理の中断時間を設定(通信条件設定後、0.5秒間中断に設定)
ec.AsclilLineOut = 1500 'タイムアウトの設定(タイムアウトを1.5秒に設定)
ec.HandShaking = "X" 'ハンドシェイクの設定(ハンドシェイクはXon/Xoffに設定)
ec.WAITms = 500 '処理の中断時間を設定(通信条件設定後、0.5秒間中断に設定)

ec.AsclilLine = "REM ON" 'リモート機能をONに設定
ec.WAITms = 4000 '処理の中断時間を設定(通信条件設定後、4秒間中断に設定)

ec.AsclilLine = "DOS" '電動ビュレットをDOSモードに変更
ec.WAITms = 500 '処理の中断時間を設定(通信条件設定後、4秒間中断に設定)

ec.AsclilLine = "F" 'シリンジへ試薬を吸引する
ec.WAITms = 60000 '処理の中断時間を設定(通信条件設定後、60秒間中断に設定)

ec.AsclilLine = "VUP 40" 'シリンジからの吐出流速を設定(吐出流速を40mL/minに設定)
ec.WAITms = 500 '処理の中断時間を設定(通信条件設定後、0.5秒間中断に設定)

ec.AsclilLine = "VDW 30" 'シリンジへの吸引流速を設定(吸引流速を30mL/minに設定)
ec.WAITms = 500 '処理の中断時間を設定(通信条件設定後、0.5秒間中断に設定)

ec.AsclilLine = "VLI 25" 'シリンジからの吐出容量を設定(吐出容量を25mLに設定)
ec.WAITms = 500 '処理の中断時間を設定(通信条件設定後、0.5秒間中断に設定)

ec.AsclilLine = "G" 'シリンジから試料を吐出
ec.WAITms = 4000 '処理の中断時間を設定(通信条件設定後、4秒間中断に設定)

```

End Sub

Figure 8 Electric burette test program

The communication setting was set according to Table 3.

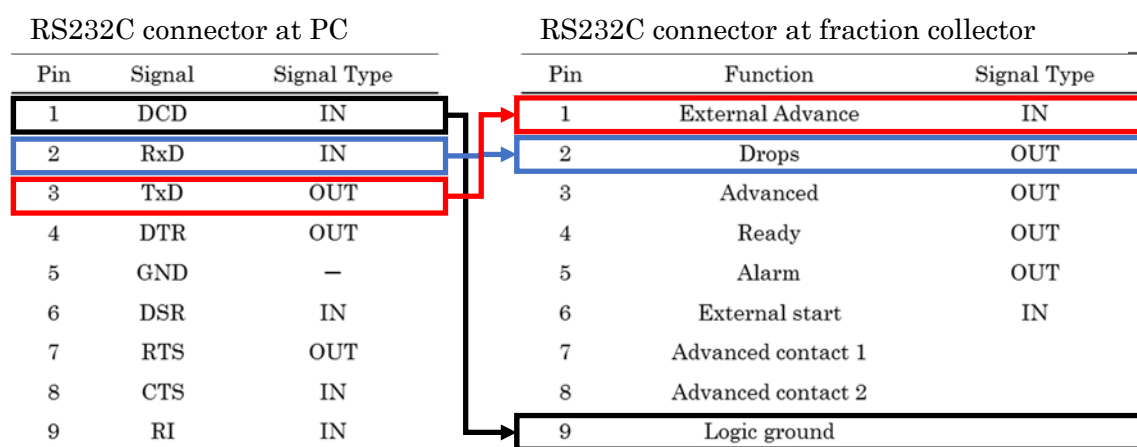


Figure 9 Cable connection of fraction collector

The RS232C connectors were used for both PC and fraction collector.

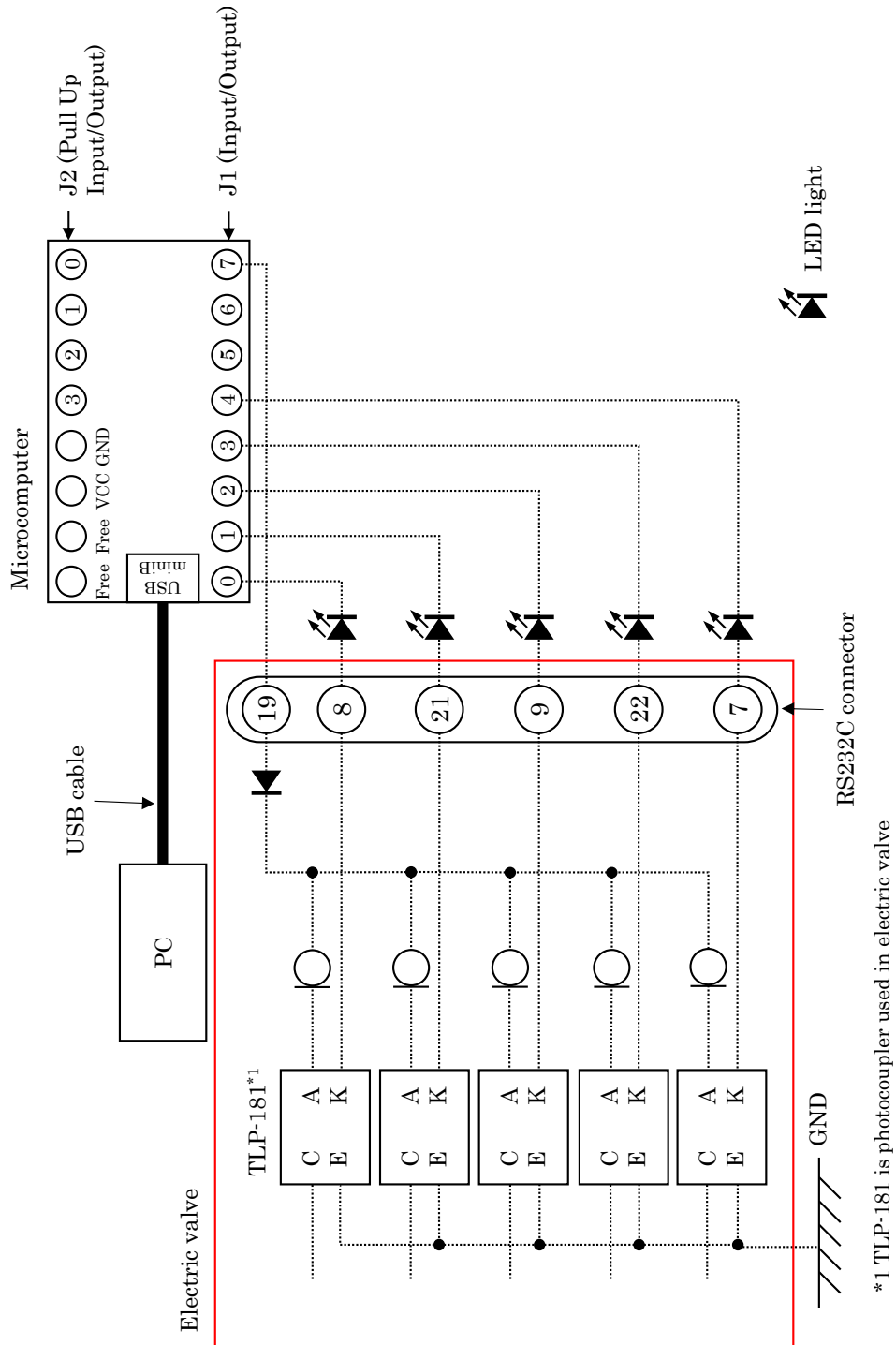


Figure 11 Cable connection of microcomputer and electric valve
The PC and microcomputer were connected by USB cable. Cables from microcomputer were connected to the RS232C connector of electric valve.

Table 6 Valve position by binary control

Decimal number was changed to hexadecimal by VBA program and sent to the microcomputer. Binary output voltages from microcomputer control the valve position of electric valve.

Pin number of RS232C at electric valve	19	---	---	7	22	9	21	8		
Pin number of microcomputer	J1-7	J1-6	J1-5	J1-4	J1-3	J1-2	J1-1	J1-0	Binary	Decimal number
Initial binary when valve is connected to microcomputer	ON	ON	ON	ON	ON	ON	ON	ON	11111111	255
Valve position	1	ON	ON	ON	OFF	ON	ON	ON	OFF	11101110 $238 = 255 - 2^4 - 2^0$
	2	ON	ON	ON	OFF	ON	ON	OFF	ON	11101101 $237 = 255 - 2^4 - 2^1$
	3	ON	ON	ON	OFF	ON	ON	OFF	OFF	11101100 $236 = 255 - 2^4 - 2^1 - 2^0$
	4	ON	ON	ON	OFF	ON	OFF	ON	ON	11101011 $235 = 255 - 2^4 - 2^2$
	5	ON	ON	ON	OFF	ON	OFF	ON	OFF	11101010 $234 = 255 - 2^4 - 2^2 - 2^0$
	6	ON	ON	ON	OFF	ON	OFF	OFF	ON	11101001 $233 = 255 - 2^4 - 2^2 - 2^1$
	7	ON	ON	ON	OFF	ON	OFF	OFF	OFF	11101000 $232 = 255 - 2^4 - 2^2 - 2^1 - 2^0$
	8	ON	ON	ON	OFF	OFF	ON	ON	ON	11100111 $231 = 255 - 2^4 - 2^3$
	9	ON	ON	ON	OFF	OFF	ON	ON	OFF	11100110 $230 = 255 - 2^4 - 2^3 - 2^0$
	10	ON	ON	ON	OFF	OFF	ON	OFF	ON	11100101 $229 = 255 - 2^4 - 2^3 - 2^1$
	11	ON	ON	ON	OFF	OFF	ON	OFF	OFF	11100100 $228 = 255 - 2^4 - 2^3 - 2^1 - 2^0$
	12	ON	ON	ON	OFF	OFF	OFF	ON	ON	11100011 $227 = 255 - 2^4 - 2^3 - 2^2$

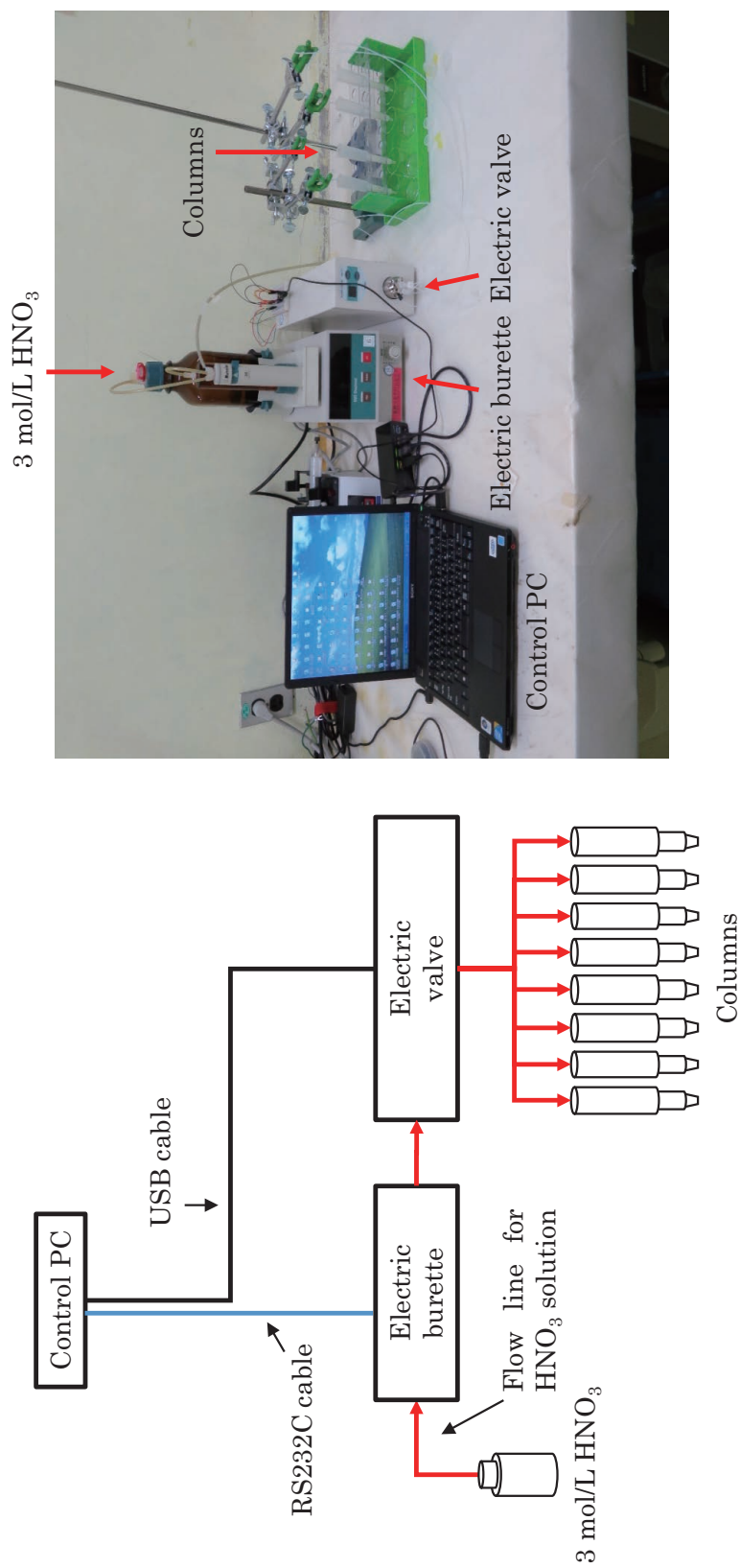


Figure 12 Schematic and photograph of automatic conditioning system

The system was composed of electric burette and electric valve. Eight columns can be simultaneously conditioned by automatic conditioning system.

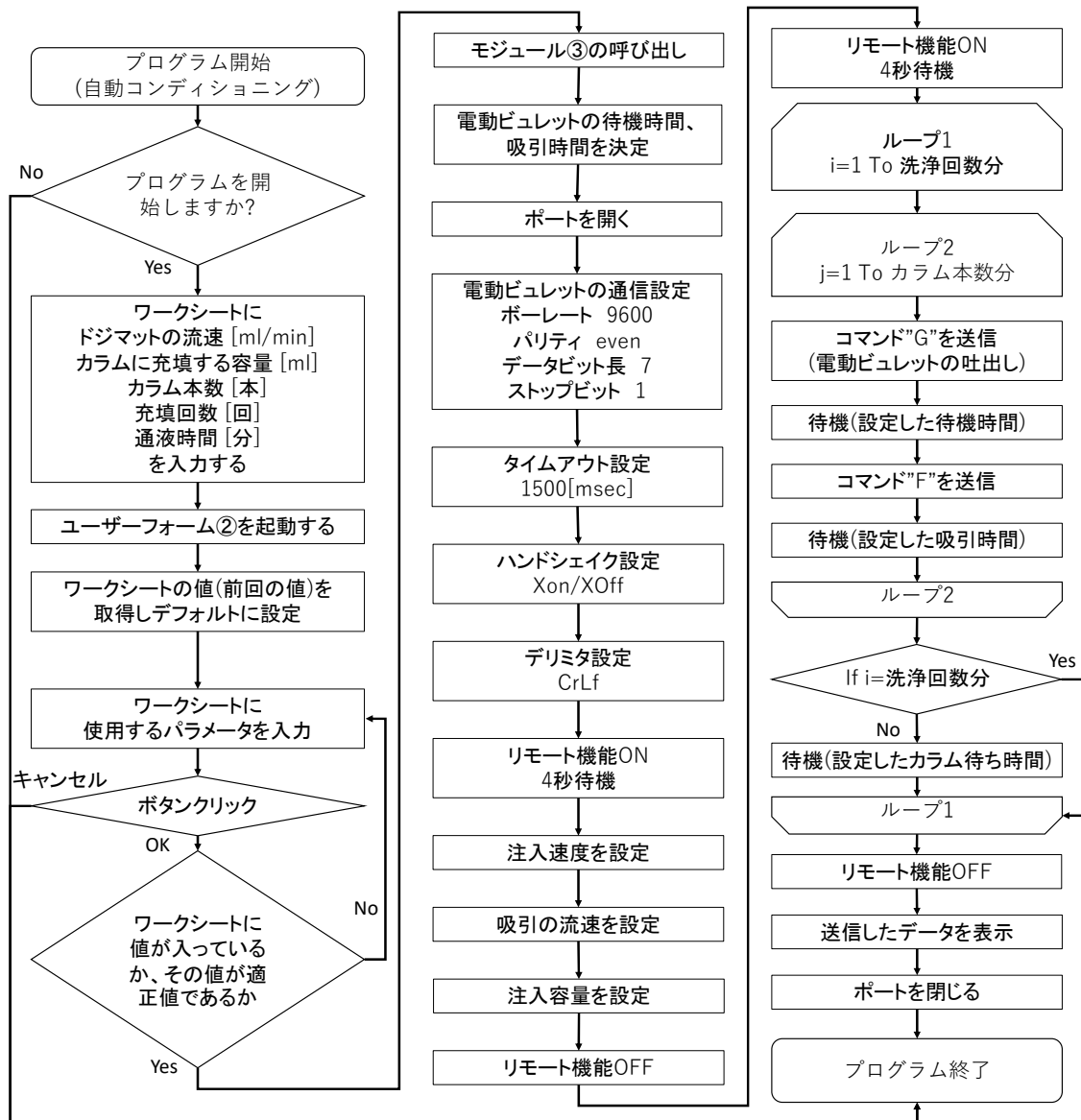


Figure 13 Program flow chart for automatic conditioning system

The program controls electric burette and electric valve for conditioning operation.

コンディショニング装置の各パラメータ

注入の流速 [ml/min]	20
カラムに充填する容量 [ml]	3
カラム数 [本]	1
充填回数 [回]	2
通液時間 [分]	10
吸引の流速 [ml/min]	60

注意)ここは 基本変更しなくても大丈夫です

コンディショニングプログラム 開始

プログラム中断

Dosimat接続テスト

Microsoft Excel

プログラムを開始しますか?

はい(Y) いいえ(N)

UserForm2

全て整数で入力してください

注入の流速 [ml/min]	20	$1 \leq x \leq 60$
1カラムの容量 [ml]	3	$1 \leq x \leq 20$
カラムの本数 [本]	1	$1 \leq x$
充填回数 [回]	2	$1 \leq x$
通液時間 [分]	10	$1 \leq x$

OK キャンセル

Figure 14 User forms made for automatic conditioning system

The flow rate of reagent injection, additional volume of reagent, number of columns, washing time, and waiting time after reagent addition were input to the user form.

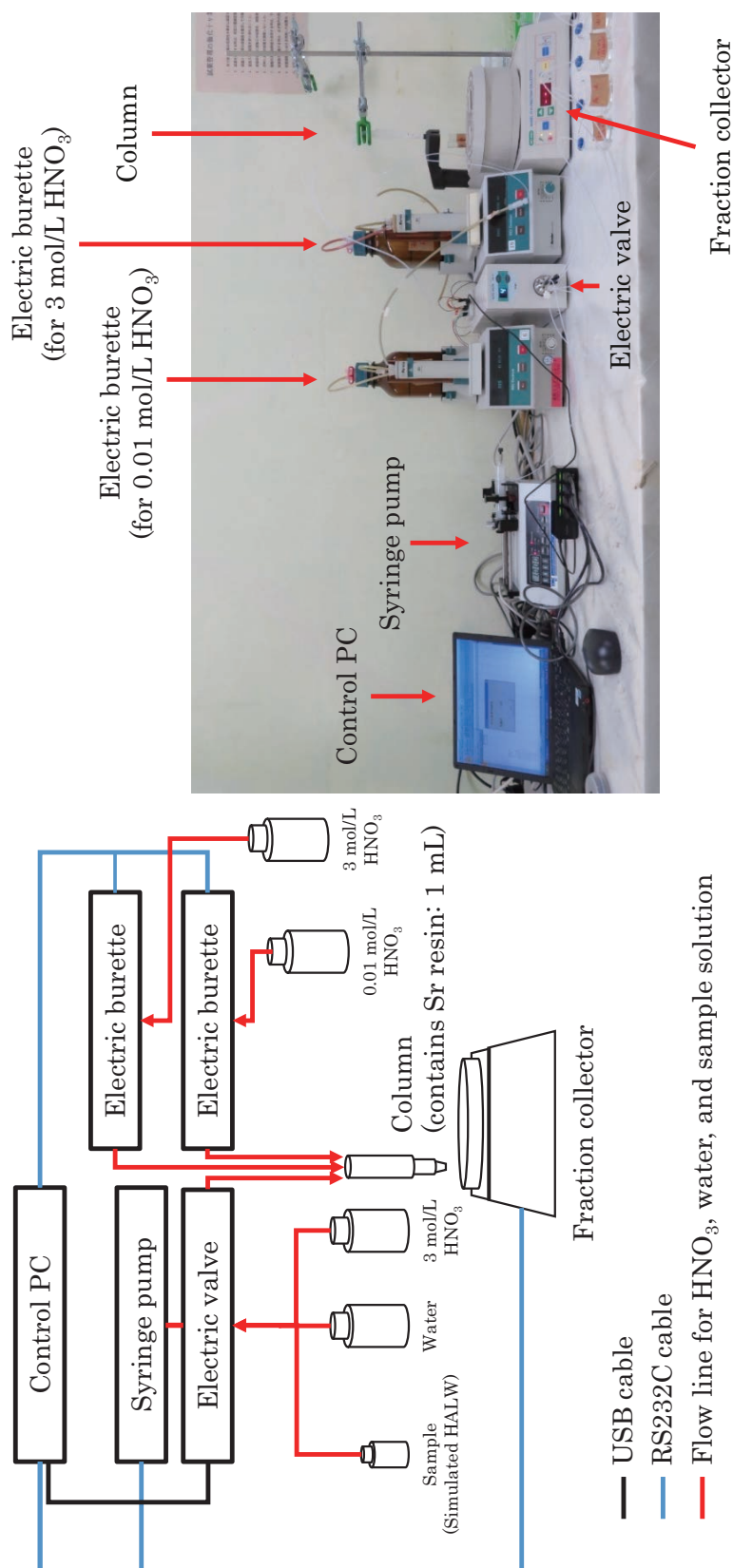


Figure 15 Schematic and photograph of automatic separation system

The system was composed of syringe pump, electric burettes, electric valve, and fraction collector. Strontium resin was filled in the column to separate Sr from SHAW.

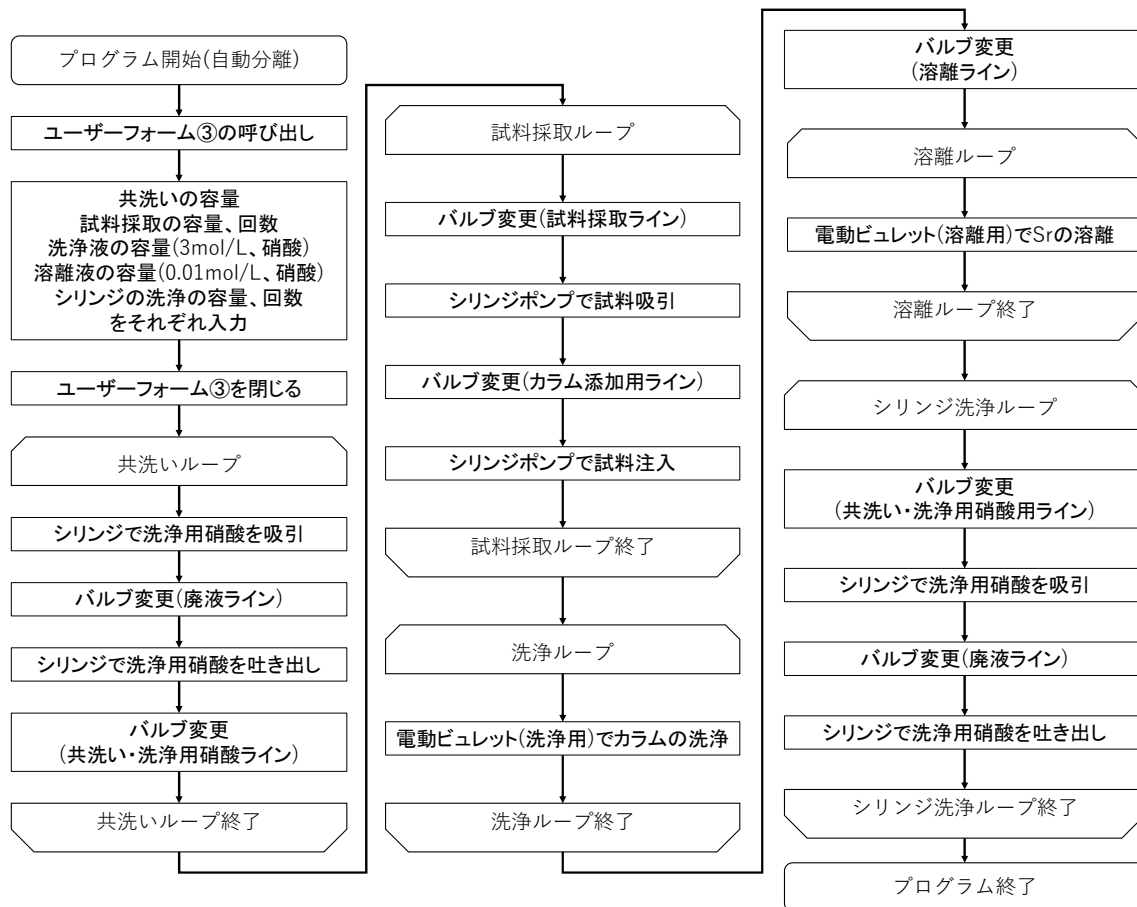


Figure 16 Program flow chart for automatic separation system

The program controls syringe pump, electric burette, fraction collector, and electric valve for Sr separation operation. The valve #1 was connected to washing solution of syringe pump, the valve #2 was connected to waste container, the valve #3 was connected to sample solution, and the valve #4 was connected to the separation column.

UserForm6

共洗いの流速	<input type="text" value="3"/>	mL/min	3M硝酸の流速	<input type="text" value="3"/>	mL/min
共洗いの容量	<input type="text" value="2"/>	mL	3M硝酸の容量	<input type="text" value="1"/>	mL
試料採取の流速	<input type="text" value="3"/>	mL/min	0.1M硝酸の流速	<input type="text" value="3"/>	mL/min
試料採取の容量	<input type="text" value="1"/>	mL	0.1M硝酸の容量	<input type="text" value="1"/>	mL
通液待ち時間		<input type="text" value="1"/>	分		

OK キャンセル

UserForm1

カラム洗浄5回目

待機中 93%

閉じる

UserForm1

溶離中2回目

待機中 52%

閉じる

Figure 17 User forms made for automatic separation system

The flow rate and volume of 3 mol/L HNO_3 solution, 0.01 mol/L HNO_3 solution, and syringe washing solution were input to the user form.

Table 7 Measurement results of strontium

The Sr in SHAW was separated and recovered by automatic and manual separation. The RSD represents relative standard deviation of the measurements (N = 3).

Method	Measurement results	
	Sr concentration (g/L)	RSD (%)
Automatic separation	1.0 ± 0.0030	0.3
Manual separation	1.2 ± 0.029	2.4
Prepared value	1.1	---

付録-1 電動バルブの制御プログラム

A1-1. はじめに

電動バルブの制御プログラムを以下に示す。なお、本プログラムは、電動バルブのバルブ位置を No.3 に変更するものである。

A1-2. プログラム

作成したプログラムはモジュール①～⑤で構成されている。モジュール①は、USB 接続機器であるマイクロコンピュータを VBA で使用するための宣言文である。モジュール②は、VBA で PC とマイクロコンピュータを接続するためのプロシージャである。モジュール③は、VBA で PC からマイクロコンピュータへコマンドを送信するためのプロシージャである。モジュール④は、電動バルブの制御プロシージャである。モジュール⑤は、バルブ位置を指定するためのプロシージャである。本プログラムは上述したように、バルブ位置を No.3 に変更するプログラムであるが、バルブ位置をその他の番号に変更するにはモジュール⑤内の「Call valve(0,1,4)」の括弧内の数値を変更すれば良い。この数値は、J1 側端子の OFF にしたいピン番号に相当する。このため、Table 6 から変更したいバルブ位置にするための J1 側端子を確認し、相当するピン番号を括弧内に入力する必要がある。

A1-2.1 モジュール① USB 接続の宣言

Public Amount As Long'モジュール全体で Amount を宣言。

```
Private Const OPEN_EXISTING = 3
Private Const INVALID_HANDLE_VALUE = -1
Private Const DIGCF_PRESENT = &H2
Private Const DIGCF_DEVICEINTERFACE = &H10
Private Const GENERIC_WRITE = &H40000000
Private Const GENERIC_READ = &H80000000
Private Const FILE_ATTRIBUTE_NORMAL = &H80
Private Const CREATE_ALWAYS = 2
Private Const OPEN_ALWAYS = 4
Private Const FILE_SHARE_READ = &H1
Private Const FILE_SHARE_WRITE = &H2
Private Const WAIT_OBJECT_0 = 0
```

```
Private Type GUID
    Data1 As Long
    Data2 As Integer
    Data3 As Integer
    Data4(7) As Byte
End Type
```

```
Private Type HIDD_ATTRIBUTES
    Size As Long
    VendorID As Integer
    ProductID As Integer
    VersionNumber As Integer
End Type
```

```
Private Type HIDP_CAPS
```

```

Usage As Integer
UsagePage As Integer
InputReportByteLength As Integer
OutputReportByteLength As Integer
FeatureReportByteLength As Integer
Reserved(16) As Integer
NumberLinkCollectionNodes As Integer
NumberInputButtonCaps As Integer
NumberInputvalueCaps As Integer
NumberInputDataIndices As Integer
NumberOutputButtonCaps As Integer
NumberOutputvalueCaps As Integer
NumberOutputDataIndices As Integer
NumberFeatureButtonCaps As Integer
NumberFeaturevalueCaps As Integer
NumberFeatureDataIndices As Integer
End Type

```

```
Private Type HidP_value_Caps
```

```

UsagePage As Integer
ReportID As Byte
IsAlias As Long
BitField As Integer
LinkCollection As Integer
LinkUsage As Integer
LinkUsagePage As Integer
IsRange As Long
IsStringRange As Long
IsDesignatorRange As Long
IsAbsolute As Long
HasNull As Long
Reserved As Byte
BitSize As Integer
ReportCount As Integer
Reserved2 As Integer
Reserved3 As Integer
Reserved4 As Integer
Reserved5 As Integer
Reserved6 As Integer
LogicalMin As Long
LogicalMax As Long
PhysicalMin As Long
PhysicalMax As Long
UsageMin As Integer
UsageMax As Integer
StringMin As Integer
StringMax As Integer
DesignatorMin As Integer
DesignatorMax As Integer
DataIndexMin As Integer
DataIndexMax As Integer

```

```
End Type
```

```
Private Type SP_DEVICE_INTERFACE_DATA
```

```

cbSize As Long
InterfaceClassGuid As GUID
Flags As Long
Reserved As Long

```

```
End Type
```

```
Private Type SP_DEVICE_INTERFACE_DETAIL_DATA
```

```

cbSize As Long
devicepath1 As Byte
devicepath2 As Byte

```

```
End Type
```

Private Type SP_DEVINFO_DATA

 cbSize As Long
 ClassGuid As GUID
 DevInst As Long
 Reserved As Long

End Type

Private Type OVERLAPPED

 Internal As Long
 InternalHigh As Long
 Offset As Long
 OffsetHigh As Long
 hEvent As Long

End Type

Private Declare Function HidD_GetHidGuid Lib "hid.dll" (ByRef HidGuid As GUID) As Long

Private Declare Function CreateFile Lib "Kernel32" Alias "CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, ByRef lpSecurityAttributes As Long, ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long

Private Declare Function ReadFile Lib "Kernel32" (ByVal hFile As Long, ByRef lpBuffer As Byte, ByVal nNumberOfBytesToRead As Long, ByRef lpNumberOfBytesRead As Long, ByRef lpOverlapped As OVERLAPPED) As Long

Private Declare Function WriteFile Lib "Kernel32" (ByVal hFile As Long, ByRef lpBuffer As Byte, ByVal nNumberOfBytesToWrite As Long, ByRef lpNumberOfBytesWritten As Long, ByRef lpOverlapped As OVERLAPPED) As Long

Private Declare Function CloseHandle Lib "Kernel32" (ByVal hObject As Long) As Long

Private Declare Function HidD_GetPreparedData Lib "hid.dll" (ByVal HidDeviceObject As Long, ByRef PreparedData As Long) As Long

Private Declare Function HidP_GetCaps Lib "hid.dll" (ByVal PreparedData As Long, ByRef Capabilities As HIDP_CAPS) As Long

Private Declare Function HidD_FreePreparedData Lib "hid.dll" (ByRef PreparedData As Long) As Long

Private Declare Function HidD_GetAttributes Lib "hid.dll" (ByVal HidDeviceObject As Long, ByRef Attributes As HIDD_ATTRIBUTES) As Long

Private Declare Function SetupDiCreateDeviceInfoList Lib "setupapi.dll" (ByRef ClassGuid As GUID, ByVal hwndParent As Long) As Long

Private Declare Function SetupDiDestroyDeviceInfoList Lib "setupapi.dll" (ByVal DeviceInfoSet As Long) As Long

Private Declare Function SetupDiEnumDeviceInterfaces Lib "setupapi.dll" (ByVal DeviceInfoSet As Long, ByVal DeviceInfoData As Long, ByRef InterfaceClassGuid As GUID, ByVal MemberIndex As Long, ByRef DeviceInterfaceData As SP_DEVICE_INTERFACE_DATA) As Long

Private Declare Function SetupDiGetClassDevs Lib "setupapi.dll" Alias "SetupDiGetClassDevsA" (ByRef ClassGuid As GUID, ByVal Enumerator As String, ByVal hwndParent As Long, ByVal Flags As Long) As Long

Private Declare Function SetupDiGetDeviceInterfaceDetail Lib "setupapi.dll" Alias "SetupDiGetDeviceInterfaceDetailW" (ByVal DeviceInfoSet As Long, ByRef DeviceInterfaceData As SP_DEVICE_INTERFACE_DATA, ByVal DeviceInterfaceDetailData As Long, ByVal DeviceInterfaceDetailDataSize As Long, ByRef RequiredSize As Long, ByVal DeviceInfoData As Long) As Long

```
Private Declare Function RtlMoveMemory Lib "Kernel32" (dest As Any, src As Any, ByVal Count As Long) As Long
```

```
Declare Function CreateEvent Lib "Kernel32" Alias "CreateEventA" (lpEventAttributes As Long, ByVal bManualReset As Long, ByVal bInitialState As Long, ByVal lpName As String) As Long
```

```
Private Declare Function GetOverlappedResult Lib "kernel32.dll" (ByVal hFile As Integer, ByRef lpOverlapped As OVERLAPPED, ByRef lpNumberOfBytesTransferred As Long, ByVal bWait As Boolean) As Integer
```

```
Declare Function WaitForSingleObject Lib "Kernel32" (ByVal hHandle As Long, ByVal dwMilliseconds As Long) As Long
```

```
Dim g_hDevice As Long
```

```
Dim g_nInputSize As Long
```

```
Dim g_nOutputSize As Long
```

```
Private Sub Destroy()
```

```
    If g_hDevice <> INVALID_HANDLE_VALUE And g_hDevice <> 0 Then  
        CloseHandle g_hDevice
```

```
    End If
```

```
    g_hDevice = INVALID_HANDLE_VALUE
```

```
    g_nInputSize = 0
```

```
    g_nOutputSize = 0
```

```
End Sub
```

A1-2.2 モジュール② マイクロコンピュータとの接続

```
Sub Connect()
```

```
    Dim nVendorID As Long
```

```
    Dim nDeviceID As Long
```

```
    Dim ret As Long
```

```
    Dim dwIndex As Long
```

```
    Dim hDevInfo As Long
```

```
    Dim sDeviceInterfaceData As SP_DEVICE_INTERFACE_DATA
```

```
    Dim DetailData As Long
```

```
    Dim sDeviceInterfaceDetailData As SP_DEVICE_INTERFACE_DETAIL_DATA
```

```
    Dim Needed As Long
```

```
    Dim strDevicePath As String
```

```
    Dim hDevice As Long
```

```
    Dim sHidpCaps As HIDP_CAPS
```

```
    Dim NumMax As Integer
```

```
    Dim guidHid As GUID
```

```
    Dim PreparedData As Long
```

```
    Dim dwCount As Long
```

```
    Dim strBuff As String
```

```
Destroy
```

```
Worksheets("Sheet1").Activate 'ベンダーID、デバイス ID を入力したワークシートをアクティブ化
```

```
strBuff = Cells(2, 1)
```

```
nVendorID = Hex2Long(strBuff) 'ベンダーID を入力(ここではワークシートの値を読み込み)
```

```
strBuff = Cells(2, 2)
```

```
nDeviceID = Hex2Long(strBuff) 'デバイス ID を入力(ここではワークシートの値を読み込み)
```

```
ret = HidD_GetHidGuid(guidHid)
```

```
hDevInfo = SetupDiGetClassDevs(guidHid, vbNullString, 0, (DIGCF_PRESENT Or DIGCF_DEVICEINTERFACE))
```

```
dwIndex = 0
```

```
sDeviceInterfaceData.cbSize = LenB(sDeviceInterfaceData)
```

```
Do While (SetupDiEnumDeviceInterfaces(hDevInfo, 0, guidHid, dwIndex, sDeviceInterfaceData))
```

```
dwIndex = dwIndex + 1
```

```
ret = SetupDiGetDeviceInterfaceDetail(hDevInfo, sDeviceInterfaceData, 0, 0, Needed, 0)
```

```
Dim DetailDataBuffer() As Byte
```

```

DetailData = Needed
sDeviceInterfaceDetailData.cbSize = Len(sDeviceInterfaceDetailData)
ReDim DetailDataBuffer(Needed)
RtlMoveMemory DetailDataBuffer(0), sDeviceInterfaceDetailData, 4
ret = SetupDiGetDeviceInterfaceDetail(hDevInfo, sDeviceInterfaceData,
VarPtr(DetailDataBuffer(0)), DetailData, Needed, 0)

If ret Then
    Dim pBuff() As Byte
    ReDim pBuff(Needed - 6)
    For i = 0 To Needed - 6
        pBuff(i) = DetailDataBuffer(i + 4)
    Next
    strDevicePath = pBuff
    hDevice = CreateFile(strDevicePath, GENERIC_READ Or GENERIC_WRITE,
(FILE_SHARE_READ Or FILE_SHARE_WRITE), 0, OPEN_EXISTING, 0, 0)
If hDevice <> INVALID_HANDLE_VALUE Then
    Dim sHiddAttributes As HIDD_ATTRIBUTES
    ret = HidD_GetAttributes(hDevice, sHiddAttributes)
If ret And sHiddAttributes.VendorID = nVendorID And sHiddAttributes.ProductID = nDeviceID
Then
    ret = HidD_GetPreparedData(hDevice, PreparedData)
If ret Then
    ret = HidP_GetCaps(PreparedData, sHidpCaps)
If ret Then
    g_nInputSize = sHidpCaps.InputReportByteLength
    g_nOutputSize = sHidpCaps.OutputReportByteLength
    For i = 0 To g_nOutputSize - 1
        Cells(5, 5 + i) = "00"
    Next
    For i = g_nOutputSize To g_nOutputSize + 5
        Cells(5, 5 + i) = ""
    Next
    For i = 0 To g_nInputSize + 5
        Cells(7, 5 + i) = ""
    Next
End If

HidD_FreePreparedData PreparedData

If ret Then
    g_hDevice = hDevice
    Cells(15, 2) = "接続に成功しました。" & vbCrLf & "PC->デバイスへのデータサイズは" &
g_nOutputSize & "バイト" & vbCrLf & "デバイス->PC へのデータサイズは" & g_nInputSize & "バ
イトです。"
    SetupDiDestroyDeviceInfoList hDevInfo
    Exit Sub
End If

End If
End If

CloseHandle hDevice
g_hDevice = INVALID_HANDLE_VALUE

End If
End If

Loop
SetupDiDestroyDeviceInfoList hDevInfo
Cells(15, 2) = "接続に失敗しました。"
End Sub

```

A1-2.3 モジュール③ マイクロコンピュータへ送信を行うプロシージャ

```

Function Send()
    Dim sOverlapped As OVERLAPPED

    If g_hDevice = INVALID_HANDLE_VALUE Or g_hDevice = 0 Then
        Exit Function
    End If

    sOverlapped.hEvent = CreateEvent(0, False, True, "")
    Dim dwWait As Long
    Dim dwWritten As Long
    Dim pBuff() As Byte
    Dim strSendedData As String
    ReDim pBuff(g_nOutputSize)

    Cells(5, 6).Value = "20" 'マイクロコンピュータのコマンド；デジタル入出力 デジタル入出力、コマ
    ンド受信後、全ピン入力を行い指示された出力を実行
    Cells(5, 7).Value = "1" 'J1 側入出力端子を設定(全て ON にする)
    Cells(5, 8).Value = Hex(Amount) '16 進数へ変換

    For i = 0 To g_nOutputSize
        pBuff(i) = Hex2Long(Cells(5, 5 + i))
        strSendedData = strSendedData & " " & Byte2Hex(pBuff(i))
    Next

    pBuff(0) = 0
    ret = WriteFile(g_hDevice, pBuff(0), g_nOutputSize, dwWritten, sOverlapped)
    ret = False
    dwWait = WaitForSingleObject(sOverlapped.hEvent, 1000)

    If dwWait = WAIT_OBJECT_0 Then
        ret = GetOverlappedResult(g_hDevice, sOverlapped, dwWritten, True)
    End If
    If ret Then
        Cells(15, 2) = "送信に成功しました。" & vbCrLf & "1 バイト目は「00」固定です。" & vbCrLf &
        strSendedData
    End If
    End If

End Function

Private Function Hex2Long(strHex As String) As Long
    Hex2Long = Val(If(Left(strHex, 1) = "&", "", "&H") & strHex)
End Function

Private Function Byte2Hex(cbData As Byte) As String
    If cbData < &H10 Then
        Byte2Hex = "0" & Hex(cbData)
    Else
        Byte2Hex = Hex(cbData)
    End If
End Function

```

A1-2.4 モジュール④ 電動バルブの制御プロシージャ

```

Function valve(ParamArray MyVal()) 'バルブユニット制御プログラム
    Dim i As Integer 'For に使用する変数 i を宣言
    Dim num(7) 'num という名前で配列型として宣言
    Sheets("Sheet1").Activate
    Amount = 255 'マイクロコンピュータに送信するには 16 進数でなければならないため、16 進数変換
    に使用する変数 Amount に 1Byte で表現できる最大の数 255(2 進数で 11111111)を設定。

    For i = 0 To 7

```

```

    num(i) = 2 ^ i 'num(0)から num(7)までに 2^i の値を代入
Next i

```

```

Call Connect

```

```

For i = LBound(MyVal) To UBound(MyVal) 'MyVal は可変引数である。引数の数だけループを行う。
例: call Valve(0,1,4)なら 3 回

```

```

    Amount = Amount - num(MyVal(i)) '2 の引数乗(2^引数)の値を 255 から引く 例:call Valve(0,1,4)な
    ら、255-1-2-16=236(2 進数で 11101100)
Next i

```

```

Call Send 'USB で送信を行うプロシージャー、Amount は Send 内で 16 進数に変換され、セルに入力
している。これを取得しデータを送信している。

```

```

End Function

```

A1-2.5 モジュール⑤ バルブ位置の指定プロシージャー

```

Sub use()

```

```

    Call valve '引数を入力しないことで 255 (11111111) を送信。USB-AKI のポート 0～7 に対応してお
    り、バルブの全てのポートが ON になる

```

```

    ec.WAITmS = 500 '待機時間 500ms

```

```

    Call valve(0, 1, 4) 'バルブ位置を 3 番に変更。valve プロシージャーに引数 0,1,4 を送り、戻り値が 236(2
    進数で 11101100)。マイクロコンピュータの J1-0 番、J1-1 番、J1-4 番ポートが OFF になり、出力が
    0V になる。変更する際は、Call valve の括弧内に OFF にしたい J1 側ピンの番号を記入する。

```

```

    ec.WAITmS = 300 'バルブ変更後の待機時間 300ms

```

```

    Call valve '再び valve プロシージャーを呼び出し、全てのポートを ON

```

```

    ec.WAITmS = 500 '待機時間 500ms

```

```

End Sub

```

付録-2 自動コンディショニング装置と自動分離装置の制御プログラム

A2-1. はじめに

自動コンディショニング装置と自動分離装置の制御プログラムをユーザーフォーム、標準モジュールの順に記載する。

A2-2. プログラム

作成したプログラムは、ユーザーフォーム①～⑥、モジュール①～⑥から構成される。ユーザーフォーム①～⑥は、自動コンディショニング装置と自動分離装置の作動に必要なパラメータ(添加試薬類の流速、容量、カラム本数等)の入力と読み取り、プログラムの開始、終了を行うためのものである。モジュール①は、シリンジポンプ用の制御プログラムである。モジュール②は、フラクションコレクター用の制御プログラムである。モジュール③は、電動ビュレットの制御プログラムである。モジュール④は、電動バルブの制御プログラムである。モジュール⑤は、コンディショニング及びカラム分離操作のため、各ユーザーフォームとモジュールを呼び出し、操作を開始させるものである。モジュール⑥は、その他の補助用に作成したプログラムである。

A2-2.1 ユーザーフォーム①

```
Private Sub CommandButton1_Click()
    Unload UserForm1 '「閉じる」ボタンを押すと UserForm1 が終了する
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    If CloseMode = 0 Then Cancel = True '×ボタンを効かないようにする
End Sub
```

A2-2.2 ユーザーフォーム②

```
Private Sub UserForm_Initialize() 'TextBox の初期値を取得しデフォルトとして設定するプログラム
    TextBox1.Value = Cells(3, 4).Value 'ワークシートの入力値を読み込み
    TextBox2.Value = Cells(4, 4).Value 'ワークシートの入力値を読み込み
    TextBox3.Value = Cells(5, 4).Value 'ワークシートの入力値を読み込み
    TextBox4.Value = Cells(6, 4).Value 'ワークシートの入力値を読み込み
    TextBox5.Value = Cells(7, 4).Value 'ワークシートの入力値を読み込み
End Sub

Private Sub CommandButton1_Click()
    '入力された値の確認プログラム、入力値が空欄又は適正範囲外の場合再度入力させる
    If TextBox1.Value = "" Then
        MsgBox "流速が入力されていません"
    ElseIf TextBox1.Value <= 0 Or TextBox1.Value > 60 Then
        MsgBox "流速は 0～60 以下のみ有効です"
    ElseIf TextBox2.Value = "" Then
        MsgBox "カラムに充填する容量が入力されていません"
    ElseIf TextBox2.Value <= 0 Or TextBox2.Value > 20 Then
        MsgBox "カラムに充填する容量は 0～20 以下のみ有効です"
    ElseIf TextBox3.Value = "" Then
        MsgBox "カラムの本数が入力されていません"
```



```

ElseIf TextBox3.Value < 1 Then
    MsgBox "カラムの本数は 1 以上のみ有効です"
ElseIf TextBox4.Value = "" Then
    MsgBox "充填回数が入力されていません"
ElseIf TextBox4.Value < 1 Then
    MsgBox "充填回数は 1 以上のみ有効です"
ElseIf TextBox5.Value = "" Then
    MsgBox "通液時間が入力されていません"
ElseIf TextBox5.Value < 0 Then
    MsgBox "通液時間は 0 以上のみ有効です"
Else '値が適正範囲内ならばそれぞれのセルに値を代入して UserForm2 を閉じる
    Cells(3, 4).Value = TextBox1.Value
    Cells(4, 4).Value = TextBox2.Value
    Cells(5, 4).Value = TextBox3.Value
    Cells(6, 4).Value = TextBox4.Value
    Cells(7, 4).Value = TextBox5.Value
    Unload UserForm2
End If
End Sub

```

A2-2.3 ユーザーフォーム③

```

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    '×ボタンが効かないようにするプログラム
    If CloseMode = 0 Then Cancel = True
End Sub

```

```

Private Sub UserForm_Initialize()
    With ComboBox1
        .AddItem "UM" 'μL/min
        .AddItem "MM" 'mL/min
        .AddItem "UH" 'μL/hour
        .AddItem "MH" 'mL/hour
    End With
    With ComboBox2
        .AddItem "UL" 'μL
        .AddItem "ML" 'mL
    End With
    Worksheets("Sheet2").Activate
    TextBox1.Value = Cells(3, 4).Value
    TextBox2.Value = Cells(5, 4).Value
    TextBox3.Value = Cells(7, 4).Value
    TextBox4.Value = Cells(8, 4).Value
    ComboBox1.Value = Cells(4, 4).Value
    ComboBox2.Value = Cells(6, 4).Value
End Sub

```

```

Private Sub CommandButton1_Click()
    If TextBox1.Value = "" Then
        MsgBox "流速が入力されていません"
    ElseIf TextBox1.Value <= 0 Then
        MsgBox "流速は 1 以上が有効です"
    ElseIf TextBox2.Value = "" Then
        MsgBox "試料採取容量が入力されていません"
    ElseIf TextBox2.Value < 1 Then
        MsgBox "試料採取容量は 1 以上が有効です"
    ElseIf TextBox3.Value = "" Then
        MsgBox "充填回数が入力されていません"
    ElseIf TextBox3.Value < 1 Then
        MsgBox "充填回数は 1 以上のみ有効です"
    ElseIf TextBox4.Value = "" Then
        MsgBox "待ち時間が入力されていません"
    ElseIf TextBox4.Value < 1 Then
        MsgBox "待ち時間は 1 以上のみ有効です"
    End If
End Sub

```

```

ElseIf ComboBox1.TEXT = "" Then
    MsgBox "流速の単位を選択してください"
ElseIf ComboBox2.TEXT = "" Then
    MsgBox "容量の単位を選択してください"
Else
    Worksheets("Sheet2").Activate
    Cells(3, 4).Value = TextBox1.Value
    Cells(5, 4).Value = TextBox2.Value
    Cells(7, 4).Value = TextBox3.Value
    Cells(8, 4).Value = TextBox4.Value
    Cells(4, 4).Value = ComboBox1.Value
    Cells(6, 4).Value = ComboBox2.Value
    Unload UserForm3 'UserForm3 を閉じる
End If
End Sub

Private Sub CommandButton2_Click()
    End 'プログラムを終了する
End Sub

```

A2-2.4 ユーザーフォーム④

```

Private Sub CommandButton1_Click() '入力値が空欄又は適正範囲外の場合は再度入力させる
    If TextBox1.Value = "" Then
        MsgBox "流速が入力されていません"
    ElseIf TextBox1.Value <= 0 Or TextBox1.Value > 150 Then
        MsgBox "流速は 0～150 のみ有効です"
    ElseIf TextBox2.Value = "" Then
        MsgBox "カラムに充填する容量が入力されていません"
    ElseIf TextBox2.Value <= 0 Or TextBox2.Value > 40 Then
        MsgBox "カラムに充填する容量は 0～40 のみ有効です"
    ElseIf TextBox3.Value = "" Then
        MsgBox "カラムの本数が入力されていません"
    ElseIf TextBox3.Value < 1 Then
        MsgBox "カラムの本数は 1 以上のみ有効です"
    ElseIf TextBox4.Value = "" Then
        MsgBox "充填回数が入力されていません"
    ElseIf TextBox4.Value < 1 Then
        MsgBox "充填回数は 1 以上のみ有効です"
    ElseIf TextBox5.Value = "" Then
        MsgBox "通液時間が入力されていません"
    ElseIf TextBox5.Value <= 0 Then
        MsgBox "通液時間は 0～のみ有効です"
    Else '値が適正範囲内ならばそれぞれのセルに値を代入して UserForm2 を閉じる
        Cells(3, 4).Value = TextBox1.Value
        Cells(4, 4).Value = TextBox2.Value
        Cells(5, 4).Value = TextBox3.Value
        Cells(6, 4).Value = TextBox4.Value
        Cells(7, 4).Value = TextBox5.Value
        Unload UserForm4
    End If
End Sub

Private Sub CommandButton2_Click() 'キャンセルボタンでプログラムが終了
    MsgBox "プログラムを終了します"
    End
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    '×ボタンが効かないようにするプログラム
    If CloseMode = 0 Then Cancel = True
End Sub

Private Sub UserForm_Initialize() 'TextBox 内の値を取得しデフォルトとして設定
    TextBox1.Value = Cells(3, 4).Value

```

```

    TextBox2.Value = Cells(4, 4).Value
    TextBox3.Value = Cells(5, 4).Value
    TextBox4.Value = Cells(6, 4).Value
    TextBox5.Value = Cells(7, 4).Value
End Sub

```

A2-2.5 ユーザーフォーム⑤

```

Public cycling_size
Private Sub CommandButton1_Click()
    If cycling_size = 40 Then
        Unload UserForm5
        UserForm4.Show (vbModal)
    ElseIf cycling_size = 20 Then
        Unload Me
        UserForm2.Show (vbModal)
    Else
        MsgBox "選択してください"
    End If
End Sub

Private Sub CommandButton2_Click()
    MsgBox "プログラムを終了します"
End Sub

Private Sub OptionButton1_Click()
    cycling_size = 40
End Sub

Private Sub OptionButton2_Click()
    cycling_size = 20
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    '×ボタンが効かないようにするプログラム
    If CloseMode = 0 Then Cancel = True
End Sub

```

A2-2.6 ユーザーフォーム⑥

```

Private Sub UserForm_Initialize()
    TextBox1.Value = Cells(4, 4).Value
    TextBox2.Value = Cells(7, 4).Value + 1
    TextBox3.Value = Cells(6, 4).Value
    TextBox4.Value = Cells(7, 4).Value
    TextBox5.Value = Cells(8, 4).Value
    TextBox6.Value = Cells(9, 4).Value
    TextBox7.Value = Cells(10, 4).Value
    TextBox8.Value = Cells(11, 4).Value
    TextBox9.Value = Cells(12, 4).Value
End Sub

Private Sub CommandButton2_Click() 'キャンセルボタンでプログラムが終了
    MsgBox "プログラムを終了します"
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    '×ボタンが効かないようにするプログラム
    If CloseMode = 0 Then Cancel = True
End Sub

```

```

Private Sub CommandButton1_Click()
    '入力値の確認プログラム、入力値が空欄又は適正範囲外の場合再度入力させる
    If TextBox1.Value = "" Then
        MsgBox "共洗いの流速が入力されていません"
    ElseIf TextBox1.Value <= 0 Or TextBox1.Value >= 16 Then
        MsgBox "流速は 1 以上 16 以下のみ有効です"
    ElseIf TextBox2.Value = "" Then
        MsgBox "共洗いの容量が入力されていません"
    ElseIf TextBox2.Value < 1 Or TextBox2.Value > 24 Then
        MsgBox "共洗いの容量は 1 以上 24 以下のみ有効です" 'シリンジのメーターを見て変更する
    ElseIf TextBox3.Value = "" Then
        MsgBox "試料採取の流速が入力されていません"
    ElseIf TextBox3.Value < 1 Or TextBox3.Value > 15 Then
        MsgBox "試料採取の流速は 1 以上 16 以下のみ有効です"
    ElseIf TextBox4.Value = "" Then
        MsgBox "試料採取の容量が入力されていません"
    ElseIf TextBox4.Value < 1 Or TextBox4.Value > 24 Then
        MsgBox "試料採取の容量は 1 以上 24 以下のみ有効です" 'シリンジメーターを見て変更する
    ElseIf TextBox5.Value = "" Then
        MsgBox "カラム洗浄の硝酸の流速が入力されていません"
    ElseIf TextBox5.Value < 1 Then
        MsgBox "カラム洗浄硝酸の流速は 1 以上のみ有効です"
    ElseIf TextBox6.Value = "" Then
        MsgBox "カラム洗浄硝酸の容量が入力されていません"
    ElseIf TextBox6.Value < 1 Or TextBox6.Value > 50 Then
        MsgBox "カラム洗浄硝酸の容量は 1 以上 50 以下のみ有効です"
    ElseIf TextBox7.Value = "" Then
        MsgBox "溶離用硝酸の流速が入力されていません"
    ElseIf TextBox7.Value < 0 Or TextBox7.Value > 20 Then
        MsgBox "溶離用硝酸の流速は 0～20 以下のみ有効です"
    ElseIf TextBox8.Value = "" Then
        MsgBox "溶離用硝酸の容量が入力されていません"
    ElseIf TextBox8.Value < 0 Then
        MsgBox "溶離用硝酸の容量は 0～のみ有効です"
    ElseIf TextBox9.Value = "" Then
        MsgBox "待ち時間が入力されていません"
    ElseIf TextBox9.Value < 0 Then
        MsgBox "待ち時間は正の値のみを入力してください"
    Else '値が適正範囲内ならばそれぞれのセルに値を代入して UserForm5 を閉じる
        washwith_rate = TextBox1.Value 'ここからユーザーフォーム用プログラム
        washwith_vol = TextBox2.Value
        sample_rate = TextBox3.Value
        sample_vol = TextBox4.Value
        HNO3_3Mrate = TextBox5.Value
        HNO3_3Mvol = TextBox6.Value
        HNO3_01Mrate = TextBox7.Value
        HNO3_01Mvol = TextBox8.Value
        separate_waittime = TextBox9.Value
        Cells(4, 4).Value = TextBox1.Value
        Cells(5, 4).Value = TextBox2.Value
        Cells(6, 4).Value = TextBox3.Value
        Cells(7, 4).Value = TextBox4.Value
        Cells(8, 4).Value = TextBox5.Value
        Cells(9, 4).Value = TextBox6.Value
        Cells(10, 4).Value = TextBox7.Value
        Cells(11, 4).Value = TextBox8.Value
        Cells(12, 4).Value = TextBox9.Value
        Unload UserForm6
    End If
End Sub

```

A2-2.7 モジュール① シリンジポンプ用プログラム

Function pump(ByVal rate_para As Integer, ByVal rate_unit As String, ByVal vol_para As Integer, ByVal vol_unit As String)

```

Dim ComPort As Byte 'ポート変数宣言
ComPort = 1 'ポート番号を記入(ここでは、ポート番号を 1 に設定)
ec.COMn = ComPort 'ポートを開く
ec.Delimiter = ec.DELIMs.Cr 'デリミタの設定
ec.Setting = "9600,n,8,1" '通信条件の設定
ec.AsciiLineTimeout = 1500 'タイムアウト設定
ec.HandShaking = "N" 'ハンドシェイクなし
ec.AsciiLine = "0 DIA 20" 'シリンジの直径(mm)を設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 RAT " & rate_para & " " & rate_unit '流速とその単位を設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 VOL " & vol_para '容量を設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 VOL " & vol_unit '容量の単位を設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 DIR WDR" 'シリンジの動作方向を吸引に設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 RUN" 'ポンプ動作開始
ec.WAITmS = ((vol_para / rate_para) * 60 + 2) * 1000 '容量、流速から待機時間を算出して待機

ec.AsciiLine = "0 DIR INF" 'シリンジの動作方向を注入に設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 RUN" 'ポンプ動作開始
ec.WAITmS = ((vol_para / rate_para) * 60 + 2) * 1000 '容量と流速から待機時間を算出して待機

ec.COMnClose = ComPort 'ポートを閉じる
End Function

```

Function pump_suikomi(ByVal suikomi_rate As Single, ByVal suikomi_unit As String, ByVal suikomi_vol As Single, ByVal suikomi_volunit As String)

```

Dim ComPort As Byte 'ポート変数宣言
ComPort = 1 'ポート番号を記入(ここでは、ポート番号を 1 に設定)
ec.COMn = ComPort 'ポートを開く
ec.Delimiter = ec.DELIMs.Cr 'デリミタの設定
ec.Setting = "9600,n,8,1" '通信条件の設定
ec.AsciiLineTimeout = 1500 'タイムアウト設定
ec.HandShaking = "N" 'ハンドシェイクなし

ec.AsciiLine = "0 DIA 20" 'シリンジの直径(mm)を設定する
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 RAT " & suikomi_rate & " " & suikomi_unit '流速とその単位を設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 VOL " & suikomi_vol '容量を設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 VOL " & suikomi_volunit '容量の単位を設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 DIR WDR" 'シリンジの動作方向を吸引に設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "0 RUN" 'ポンプ動作開始

```

```

ec.WAITmS = ((suikomi_vol / suikomi_rate) * 60 + 2) * 1000 '容量と流速から待機時間を算出して待機

ec.COMnClose = ComPort 'ポートを閉じる
End Function

Function pump_hakidasi(ByVal hakidasi_rate As Single, ByVal hakidasi_unit As String, ByVal
hakidasi_vol As Single, ByVal hakidasi_volunit As String)

    Dim ComPort As Byte 'ポート変数宣言
    ComPort = 1 'ポート番号を記入(ここでは、ポート番号を 1 に設定)
    ec.COMn = ComPort 'ポートを開く
    ec.Delimiter = ec.DELIMs.Cr 'デリミタの設定
    ec.Setting = "9600,n,8,1" '通信条件の設定
    ec.ASCIIlineTimeOut = 1500 'タイムアウト設定
    ec.HandShaking = "N" 'ハンドシェイクなし

    ec.ASCIIline = "0 DIA 20" 'シリンジの直径(mm)を設定する
    ec.WAITmS = 500 '待機時間 500ms

    ec.ASCIIline = "0 VOL " & hakidasi_volunit '容量の単位を設定
    ec.WAITmS = 500 '待機時間 500ms

    ec.ASCIIline = "0 RAT " & hakidasi_rate & " " & hakidasi_unit '流速とその単位を設定
    ec.WAITmS = 500 '待機時間 500ms

    ec.ASCIIline = "0 VOL " & hakidasi_vol '容量を設定
    ec.WAITmS = 500 '待機時間 500ms

    ec.ASCIIline = "0 DIR INF" 'シリンジの動作方向を吸引に設定
    ec.WAITmS = 500 '待機時間 500ms

    ec.ASCIIline = "0 RUN" 'ポンプ動作開始
    ec.WAITmS = ((hakidasi_vol / hakidasi_rate) * 60 + 2) * 1000 '容量と流速から待機時間を算出し待機

    ec.COMnClose = ComPort 'ポートを閉じる
End Function

```

A2-2.8 モジュール② フラクションコレクター用プログラム

```

Function advance()
    ec.COMn = 1 'ポート番号の設定(ここではポート番号を 1 に設定)
    ec.WAITmS = 100 '待機時間 100ms
    ec.ASCIIline = "oooooooooooooooooooooooooooooooooooooooooooooooooooo" 'コマンドの送信
    ec.WAITmS = 1000 '待機時間 1000ms
End Function

```

A2-2.9 モジュール③ 電動ビュレット用プログラム

```

Dim sentence(4) As String
Public dosi1rate As Single
Public dosi1vol As Single
Public column_num As Long
Public column_freq As Long
Public wait_time As Single
Public fill_time As Single
Public column_time As Single
Public dosidwrate As Single

Function inputcell() '電動ビュレットへのコンディショニング用パラメータの読み取り
    Worksheets("Sheet1").Activate
    Cells(2, 3) = "コンディショニング装置の各パラメータ" 'ワークシートからパラメータの読み取り
    sentence(0) = "注入の流速 [ml/min]"

```

```

sentence(1) = "カラムに充填する容量 [ml]"
sentence(2) = "カラム数 [本]"
sentence(3) = "充填回数 [回]"
sentence(4) = "通液時間 [分]"
For i = 0 To 4
    Cells(i + 3, 3) = sentence(i)
Next i
End Function

Function dositest() '電動ビュレットの通信テスト
    Dim ComPort As Byte 'ポート変数宣言
    ComPort = 1 'ポート番号を設定(ここでは 1 に設定)
    ec.COMn = ComPort 'ポートを開く
    ec.Setting = "9600,e,7,1" '通信条件の設定
    ec.WAITmS = 500 '待機時間 500ms
    ec.AsciiLineTimeOut = 1500 'タイムアウトを 1500ms に設定
    ec.WAITmS = 500 '待機時間 500ms

    ec.HandShaking = "X" 'ハンドシェイク XOn/XOff
    ec.WAITmS = 500 '待機時間 500ms
    ec.Delimiter = ec.DELIMs.CrLf 'デリミタに CrLf を設定
    ec.WAITmS = 500 '待機時間 5000ms

    ec.AsciiLine = "REM ON" 'リモート機能を ON
    ec.WAITmS = 4000 '待機時間 4000ms

    ec.AsciiLine = "DIR" '電動ビュレットを DIR モードに変更
    ec.WAITmS = 2000 '待機時間 2000ms

    ec.AsciiLine = "DOS" '電動ビュレットを DOS モードに変更
    ec.WAITmS = 2000 '待機時間 2000ms

    ec.AsciiLine = "REM OFF" 'リモート機能を OFF
    ec.WAITmS = 4000 '待機時間 4000ms

    a$ = ec.Ascii
    MsgBox a$

    ec.InBufferClear
    ec.COMnClose = ComPort
End
End Function

Function dosimat() '電動ビュレットによるコンディショニング用プログラム
    Worksheets("Sheet1").Activate 'ワークシートをアクティブ化
    dosilrate = Cells(3, 4).Value 'ワークシートの値を dosilrate に代入
    dosilvol = Cells(4, 4).Value 'ワークシートの値を dosilvol に代入
    dosidwrate = Cells(9, 4).Value 'ワークシートの値を dosidwrate に代入
    column_num = Cells(5, 4).Value 'ワークシートの値を column_num に代入
    column_freq = Cells(6, 4).Value 'ワークシートの値を column_freq に代入
    wait_time = (dosilvol / dosilrate) * 60 + 1 '容量と流速から試薬類の注入に必要な時間を設定
    fill_time = (dosilvol / dosidwrate) * 60 + 1 '容量と流速から試薬類の吸引に必要な時間を設定
    column_time = Cells(7, 4).Value 'ワークシートから待機時間を取得
    column_time = column_time * 60

    Dim i
    Dim ComPort As Byte 'ポート変数宣言
    ComPort = 1 'ポート番号を設定(ここでは 1 に設定)
    ec.COMn = ComPort 'ポートを開く
    ec.Setting = "9600,e,7,1" '通信条件の設定
    ec.WAITmS = 500 '待機時間 500ms
    ec.AsciiLineTimeOut = 1500 'タイムアウトを 1500ms に設定
    ec.WAITmS = 500 '待機時間 500ms
    ec.HandShaking = "X" 'ハンドシェイクを XOn/XOff に設定
    ec.WAITmS = 500 '待機時間 500ms

```



```

ec.Delimiter = ec.DELIMs.CrLf 'デリミタを CrLf に設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "REM ON" 'リモート機能を ON
ec.WAITmS = 4000 '待機時間 4000ms

ec.AsciiLine = "VUP " & dosilrate 'シリンジの注入流速を dosilrate の値に設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "VDW " & dosidwrate 'シリンジへの吸引流速を dosidwrate の値に設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "VLI " & dosilvol '注入容量を dsilvol の値に設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "REM OFF" 'リモート機能を OFF
ec.WAITmS = 4000 '待機時間 4000ms

ec.AsciiLine = "REM ON" 'リモート機能を ON
ec.WAITmS = 4000 '待機時間 4000ms

For i = 1 To column_freq 'カラム充填回数を column_freq の値に設定
For j = 1 To column_num 'カラム充填本数を column_num の値に設定

ec.Ascii = "G" '試薬類の注入コマンドを送信
UserForm1.Label1.Caption = j & "本目" & i & "回目の注入です" 'UserForm1 の編集
UserForm1.Show (vbModeless) 'UserForm1 を表示
Call wait(wait_time) '待機時間として wait_time の値だけ待機
Unload UserForm1 'UserForm1 を非表示
ec.Ascii = "F" '試薬類の吸引コマンドを送信
UserForm1.Label1.Caption = j & "本目" & i & "回目の吸引です" 'UserForm1 の編集
UserForm1.Show (vbModeless) 'UserForm1 を表示
Call wait(fill_time) '待機時間として fill_time の値だけ待機
Unload UserForm1 'UserForm1 を非表示

If j < column_num Then
    Call valve(4)
    Sleep (300)
    Call valve
    Sleep (300)
End If
Next j

If i < column_freq Then 'column_freq までの回数繰り返す
    UserForm1.Label1.Caption = i & "回目の通液の待ち時間です。" & vbCrLf & "待ち時間は" &
        column_time / 60 & "分です" 'UserForm1 の待ち時間用のラベルを編集
    UserForm1.Show (vbModeless) 'UserForm1 を表示

    If column_num <> 1 Then
        Call valve(4, 0)
        ec.WAITmS = 500 '待機時間 500ms
        Call valve
    End If

    Call wait_min(column_time) '待機時間として column_time の値だけ待機
    Unload UserForm1 'UserForm1 を非表示
End If

Next i

If column_num <> 1 Then
    Call valve(0, 4)
    ec.WAITmS = 500 '待機時間 500ms
    Call valve
    ec.WAITmS = 500 '待機時間 500ms

```



```

End If

ec.AsciiLine = "REM OFF" 'リモート機能の OFF
ec.WAITmS = 4000 '待機時間 4000ms

ec.COMnClose = ComPort 'ポートを閉じる
MsgBox "コンディショニング完了"
ec.WAITmS = 500 '待機時間 500ms
End Function

Function dosimat_separate(ByVal separate As Single, ByVal sepadwrate As Single, ByVal sepavol As Single) '電動ビュレットによるカラム分離用プログラム
ec.AsciiLine = "VUP " & separate '試薬類の注入流速を separate の値に設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "VDW " & sepadwrate '試薬類の吸引流速を sepadwrate の値に設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "VLI " & sepavol '容量を sepavol の値に設定
ec.WAITmS = 100

ec.Ascii = "G" '試薬類を注入
ec.WAITmS = 1000 * sepavol * 60 / separate + 1000 '容量と流速から待機時間を求めて待機

ec.Ascii = "F" '試薬類の吸引
ec.WAITmS = 1000 * sepavol * 60 / sepadwrate + 1000 '容量と流速から待機時間を求めて待機
End Function

```

A2-2.10 モジュール④ 電動バルブ用プログラム

```

Public Amount As Long 'USB 接続の宣言

Private Const OPEN_EXISTING = 3
Private Const INVALID_HANDLE_VALUE = -1
Private Const DIGCF_PRESENT = &H2
Private Const DIGCF_DEVICEINTERFACE = &H10
Private Const GENERIC_WRITE = &H40000000
Private Const GENERIC_READ = &H80000000
Private Const FILE_ATTRIBUTE_NORMAL = &H80
Private Const CREATE_ALWAYS = 2
Private Const OPEN_ALWAYS = 4
Private Const FILE_SHARE_READ = &H1
Private Const FILE_SHARE_WRITE = &H2
Private Const WAIT_OBJECT_0 = 0

Private Type GUID
    Data1 As Long
    Data2 As Integer
    Data3 As Integer
    Data4(7) As Byte
End Type

Private Type HIDD_ATTRIBUTES
    Size As Long
    VendorID As Integer
    ProductID As Integer
    VersionNumber As Integer
End Type

Private Type HIDP_CAPS
    Usage As Integer
    UsagePage As Integer
    InputReportByteLength As Integer
    OutputReportByteLength As Integer

```

```

FeatureReportByteLength As Integer
Reserved(16) As Integer
NumberLinkCollectionNodes As Integer
NumberInputButtonCaps As Integer
NumberInputvalueCaps As Integer
NumberInputDataIndices As Integer
NumberOutputButtonCaps As Integer
NumberOutputvalueCaps As Integer
NumberOutputDataIndices As Integer
NumberFeatureButtonCaps As Integer
NumberFeaturevalueCaps As Integer
NumberFeatureDataIndices As Integer
End Type

```

```
Private Type HidP_value_Caps
```

```

UsagePage As Integer
ReportID As Byte
IsAlias As Long
BitField As Integer
LinkCollection As Integer
LinkUsage As Integer
LinkUsagePage As Integer
IsRange As Long
IsStringRange As Long
IsDesignatorRange As Long
IsAbsolute As Long
HasNull As Long
Reserved As Byte
BitSize As Integer
ReportCount As Integer
Reserved2 As Integer
Reserved3 As Integer
Reserved4 As Integer
Reserved5 As Integer
Reserved6 As Integer
LogicalMin As Long
LogicalMax As Long
PhysicalMin As Long
PhysicalMax As Long
UsageMin As Integer
UsageMax As Integer
StringMin As Integer
StringMax As Integer
DesignatorMin As Integer
DesignatorMax As Integer
DataIndexMin As Integer
DataIndexMax As Integer

```

```
End Type
```

```
Private Type SP_DEVICE_INTERFACE_DATA
```

```

cbSize As Long
InterfaceClassGuid As GUID
Flags As Long
Reserved As Long

```

```
End Type
```

```
Private Type SP_DEVICE_INTERFACE_DETAIL_DATA
```

```

cbSize As Long
devicepath1 As Byte
devicepath2 As Byte

```

```
End Type
```

```
Private Type SP_DEVINFO_DATA
```

```

cbSize As Long
ClassGuid As GUID

```

DevInst As Long
 Reserved As Long
 End Type

Private Type OVERLAPPED
 Internal As Long
 InternalHigh As Long
 Offset As Long
 OffsetHigh As Long
 hEvent As Long
 End Type

Private Declare Function HidD_GetHidGuid Lib "hid.dll" (ByRef HidGuid As GUID) As Long

Private Declare Function CreateFile Lib "Kernel32" Alias "CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, ByRef lpSecurityAttributes As Long, ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long

Private Declare Function ReadFile Lib "Kernel32" (ByVal hFile As Long, ByRef lpBuffer As Byte, ByVal nNumberOfBytesToRead As Long, ByRef lpNumberOfBytesRead As Long, ByRef lpOverlapped As OVERLAPPED) As Long

Private Declare Function WriteFile Lib "Kernel32" (ByVal hFile As Long, ByRef lpBuffer As Byte, ByVal nNumberOfBytesToWrite As Long, ByRef lpNumberOfBytesWritten As Long, ByRef lpOverlapped As OVERLAPPED) As Long

Private Declare Function CloseHandle Lib "Kernel32" (ByVal hObject As Long) As Long

Private Declare Function HidD_GetPreparedData Lib "hid.dll" (ByVal HidDeviceObject As Long, ByRef PreparedData As Long) As Long

Private Declare Function HidP_GetCaps Lib "hid.dll" (ByVal PreparedData As Long, ByRef Capabilities As HIDD_CAPS) As Long

Private Declare Function HidD_FreePreparedData Lib "hid.dll" (ByRef PreparedData As Long) As Long

Private Declare Function HidD_GetAttributes Lib "hid.dll" (ByVal HidDeviceObject As Long, ByRef Attributes As HIDD_ATTRIBUTES) As Long

Private Declare Function SetupDiCreateDeviceInfoList Lib "setupapi.dll" (ByRef ClassGuid As GUID, ByVal hwndParent As Long) As Long

Private Declare Function SetupDiDestroyDeviceInfoList Lib "setupapi.dll" (ByVal DeviceInfoSet As Long) As Long

Private Declare Function SetupDiEnumDeviceInterfaces Lib "setupapi.dll" (ByVal DeviceInfoSet As Long, ByVal DeviceInfoData As Long, ByRef InterfaceClassGuid As GUID, ByVal MemberIndex As Long, ByRef DeviceInterfaceData As SP_DEVICE_INTERFACE_DATA) As Long

Private Declare Function SetupDiGetClassDevs Lib "setupapi.dll" Alias "SetupDiGetClassDevsA" (ByRef ClassGuid As GUID, ByVal Enumerator As String, ByVal hwndParent As Long, ByVal Flags As Long) As Long

Private Declare Function SetupDiGetDeviceInterfaceDetail Lib "setupapi.dll" Alias "SetupDiGetDeviceInterfaceDetailW" (ByVal DeviceInfoSet As Long, ByRef DeviceInterfaceData As SP_DEVICE_INTERFACE_DATA, ByVal DeviceInterfaceDetailData As Long, ByVal DeviceInterfaceDetailDataSize As Long, ByRef RequiredSize As Long, ByVal DeviceInfoData As Long) As Long

Private Declare Function RtlMoveMemory Lib "Kernel32" (dest As Any, src As Any, ByVal Count As Long) As Long

Declare Function CreateEvent Lib "Kernel32" Alias "CreateEventA" (lpEventAttributes As Long,

```
ByVal bManualReset As Long, ByVal bInitialState As Long, ByVal lpName As String) As Long
```

```
Private Declare Function GetOverlappedResult Lib "kernel32.dll" (ByVal hFile As Integer, ByRef  
lpOverlapped As OVERLAPPED, ByRef lpNumberOfBytesTransferred As Long, ByVal bWait As  
Boolean) As Integer
```

```
Declare Function WaitForSingleObject Lib "Kernel32" (ByVal hHandle As Long, ByVal dwMilliseconds  
As Long) As Long
```

```
Dim g_hDevice As Long  
Dim g_nInputSize As Long  
Dim g_nOutputSize As Long
```

```
Function valve(ParamArray MyVal()) '電動バルブの制御プロシージャ  
    Dim i As Integer 'For に使用する変数 i を宣言  
    Dim num(7) 'num という名前で配列型として宣言  
    Sheets("Sheet1").Activate  
    Amount = 255 '16 進数変換に使用する変数 Amount に 1Byte で表現できる最大の数 255 を設定
```

```
    For i = 0 To 7  
        num(i) = 2 ^ i 'num(0)から num(7)までに 2^i の値を代入  
    Next i
```

```
    Call Connect
```

```
    For i = LBound(MyVal) To UBound(MyVal) 'MyVal は可変引数であり、引数の数だけループを行う  
        Amount = Amount - num(MyVal(i)) '2 の引数乗(2^引数)の値を 255 から引く  
    Next i
```

```
    Call Send 'USB で送信を行うプロシージャ
```

```
End Function
```

```
Private Sub Destroy()  
    If g_hDevice <> INVALID_HANDLE_VALUE And g_hDevice <> 0 Then  
        CloseHandle g_hDevice  
    End If  
    g_hDevice = INVALID_HANDLE_VALUE  
    g_nInputSize = 0  
    g_nOutputSize = 0  
End Sub
```

```
Sub Connect()  
    Dim nVendorID As Long  
    Dim nDeviceID As Long  
    Dim ret As Long  
    Dim dwIndex As Long  
    Dim hDevInfo As Long  
    Dim sDeviceInterfaceData As SP_DEVICE_INTERFACE_DATA  
    Dim DetailData As Long  
    Dim sDeviceInterfaceDetailData As SP_DEVICE_INTERFACE_DETAIL_DATA  
    Dim Needed As Long  
    Dim strDevicePath As String  
    Dim hDevice As Long  
    Dim sHidpCaps As HIDP_CAPS  
    Dim NumMax As Integer  
    Dim guidHid As GUID  
    Dim PreparedData As Long  
    Dim dwCount As Long  
    Dim strBuff As String
```

```
Destroy  
    Worksheets("Sheet1").Activate 'ワークシートをアクティブ化  
    strBuff = Cells(2, 1)  
    nVendorID = Hex2Long(strBuff) 'ワークシートから Vendor ID を読み込み
```

```

strBuff = Cells(2, 2)
nDeviceID = Hex2Long(strBuff) 'ワークシートから Device ID を読み込み
ret = HidD_GetHidGuid(guidHid)
hDevInfo = SetupDiGetClassDevs(guidHid, vbNullString, 0, (DIGCF_PRESENT Or DIGCF_DEVICEINTERFACE))
dwIndex = 0
sDeviceInterfaceData.cbSize = LenB(sDeviceInterfaceData)

Do While (SetupDiEnumDeviceInterfaces(hDevInfo, 0, guidHid, dwIndex, sDeviceInterfaceData))
    dwIndex = dwIndex + 1
    ret = SetupDiGetDeviceInterfaceDetail(hDevInfo, sDeviceInterfaceData, 0, 0, Needed, 0)
    Dim DetailDataBuffer() As Byte
    DetailData = Needed
    sDeviceInterfaceDetailData.cbSize = Len(sDeviceInterfaceDetailData)
    ReDim DetailDataBuffer(Needed)
    RtlMoveMemory DetailDataBuffer(0), sDeviceInterfaceDetailData, 4
    ret = SetupDiGetDeviceInterfaceDetail(hDevInfo, sDeviceInterfaceData, VarPtr(DetailDataBuffer(0)), DetailData, Needed, 0)

    If ret Then
        Dim pBuff() As Byte
        ReDim pBuff(Needed - 6)

        For i = 0 To Needed - 6
            pBuff(i) = DetailDataBuffer(i + 4)
        Next

        strDevicePath = pBuff

        hDevice = CreateFile(strDevicePath, GENERIC_READ Or GENERIC_WRITE, (FILE_SHARE_READ Or FILE_SHARE_WRITE), 0, OPEN_EXISTING, 0, 0)

        If hDevice <> INVALID_HANDLE_VALUE Then
            Dim sHiddAttributes As HIDD_ATTRIBUTES
            ret = HidD_GetAttributes(hDevice, sHiddAttributes)

            If ret And sHiddAttributes.VendorID = nVendorID And sHiddAttributes.ProductID = nDeviceID Then
                ret = HidD_GetPreparsedData(hDevice, PreparsedData)

                If ret Then
                    ret = HidP_GetCaps(PreparsedData, sHidpCaps)

                    If ret Then
                        g_nInputSize = sHidpCaps.InputReportByteLength
                        g_nOutputSize = sHidpCaps.OutputReportByteLength

                        For i = 0 To g_nOutputSize - 1
                            Cells(5, 5 + i) = "00"
                        Next

                        For i = g_nOutputSize To g_nOutputSize + 5
                            Cells(5, 5 + i) = ""
                        Next

                        For i = 0 To g_nInputSize + 5
                            Cells(7, 5 + i) = ""
                        Next
                    End If

                    HidD_FreePreparsedData PreparsedData

                    If ret Then
                        g_hDevice = hDevice
                        Cells(15, 2) = "接続に成功しました。" & vbCrLf & "PC->デバイスへのデータサイズは" &

```

```

g_nOutputSize & "バイト" & vbCrLf & "デバイス->PC へのデータサイズは" & g_nInputSize & "バ
イトです。"

SetupDiDestroyDeviceInfoList hDevInfo
Exit Sub
End If

End If
End If
CloseHandle hDevice
g_hDevice = INVALID_HANDLE_VALUE
End If
End If
Loop
SetupDiDestroyDeviceInfoList hDevInfo
Cells(15, 2) = "接続に失敗しました。"
End Sub

Function Send()
Dim sOverlapped As OVERLAPPED

If g_hDevice = INVALID_HANDLE_VALUE Or g_hDevice = 0 Then
Exit Function
End If

sOverlapped.hEvent = CreateEvent(0, False, True, "")
Dim dwWait As Long
Dim dwWritten As Long
Dim pBuff() As Byte
Dim strSendedData As String
ReDim pBuff(g_nOutputSize)

Cells(5, 6).Value = "20" 'マイクコンピュータコマンド: 全ピン入力後、指示された出力を実行
Cells(5, 7).Value = "1" 'J1 側入出力端子を設定(全て ON にする)
Cells(5, 8).Value = Hex(Amount) '16 進数へ変換

For i = 0 To g_nOutputSize
pBuff(i) = Hex2Long(Cells(5, 5 + i))
strSendedData = strSendedData & " " & Byte2Hex(pBuff(i))
Next

pBuff(0) = 0
ret = WriteFile(g_hDevice, pBuff(0), g_nOutputSize, dwWritten, sOverlapped)
ret = False
dwWait = WaitForSingleObject(sOverlapped.hEvent, 1000)

If dwWait = WAIT_OBJECT_0 Then
ret = GetOverlappedResult(g_hDevice, sOverlapped, dwWritten, True)

If ret Then
Cells(15, 2) = "送信に成功しました。" & vbCrLf & "1 バイト目は「00」固定です。" & vbCrLf &
strSendedData

End If
End If
End Function

Private Function Hex2Long(strHex As String) As Long
Hex2Long = Val(If(Left(strHex, 1) = "&", "", "&H") & strHex)
End Function

Private Function Byte2Hex(cbData As Byte) As String
If cbData < &H10 Then
Byte2Hex = "0" & Hex(cbData)
Else

```

```

        Byte2Hex = Hex(cbData)
    End If
End Function

```

A2-2.11 モジュール⑤ コンディショニング、分離操作プログラム

```

Public washwith_rate    '共洗いに使用する流速の変数
Public washwith_vol     '共洗いに使用する容量の変数
Public sample_rate      '試料採取の流速の変数
Public sample_vol       '試料採取の容量の変数
Public HNO3_3Mrate      'カラム洗浄の硝酸の流速変数
Public HNO3_3Mvol       'カラム洗浄の硝酸の容量変数
Public HNO3_01Mrate     '溶離の硝酸の流速変数
Public HNO3_01Mvol      '溶離の硝酸の容量変数
Public separate_waittime '待ち時間の変数

Sub program0 '開始ボタンを押すことでコンディショニングのプログラムが開始
    Dim msg As Integer
    msg = MsgBox("プログラムを開始しますか?", vbYesNo)

    If msg = vbNo Then
        MsgBox "プログラムを終了します"
        End
    End If

    Call inputcell 'モジュール③の関数 inputcell の呼び出し
    UserForm5.Show (vbModal) 'UserForm5 の呼び出し
    Call Connect
    Call valve
    Call Record(0)
    Call dosimat 'モジュール③の関数 dosimat の呼び出し
End Sub

Sub separate_program0 '共洗いと試料採取・洗浄のプロシーチャー
    Call main
    Worksheets("Sheet2").Activate
    UserForm6.Show (vbModal)
    Call Record(1)

    separate_waittime = separate_waittime * 60
    ec.WAITmS = 100

    Call valve(0, 4) 'バルブ位置を No.1 に変更(洗浄用硝酸)
    ec.WAITmS = 300

    Call valve
    ec.WAITmS = 300

    UserForm1.Label1.Caption = "共洗いです" 'UserForm1 を編集
    UserForm1.Show (vbModeless) 'UserForm1 を表示

    Call pump_suikomi(washwith_rate, MM, washwith_vol, ML) 'シリンジの共洗いを開始
    Call valve(1, 4) 'バルブ位置を No.2 に変更(廃液入れ)
    ec.WAITmS = 300

    Call valve
    Call pump_hakidasi(washwith_rate, MM, washwith_vol, ML) 'シリンジの共洗いを終了
    ec.WAITmS = 100

    UserForm1.Label1.Caption = "試料採取中です" 'UserForm1 を編集
    Call valve(0, 1, 4) 'バルブ位置を No.3 に変更(Sr 試料)
    ec.WAITmS = 300

    Call valve

```

```

Call pump_suikomi(sample_rate, MM, sample_vol, ML)
ec.WAITmS = 100

Call valve(2, 4) 'バルブ位置を No.4 に変更(カラム)
ec.WAITmS = 300

Call valve
Call pump_hakidasi(sample_rate, MM, sample_vol, ML) '試料採取終了後、カラムに滴下
UserForm1.Label1.Caption = "採取完了 待ち時間" 'UserForm1 を編集

Call wait_min(separate_waittime) '試料通液の間、操作を待機

For i = 1 To 10 'カラム洗浄の開始
    UserForm1.Label1.Caption = "カラム洗浄中" & i & "回目"
    UserForm1.Show (vbModeless)
    ec.COMn = 1 '3M 硝酸添加用電動ビュレットのポート番号を設定
    Call dosimat_separate(HNO3_3Mrate, HNO3_3Mrate, HNO3_3Mvol) '流速と容量を設定
    Call wait_min(separate_waittime) '洗浄液通液の間、操作を待機
Next i 'カラム洗浄の終了

Unload UserForm1
Call advance

For i = 1 To 10 'Sr の溶離
    UserForm1.Label1.Caption = "溶離中" & i & "回目"
    UserForm1.Show (vbModeless)
    ec.COMn = 1 '0.01M 硝酸用電動ビュレットのポート番号を設定
    Call dosimat_separate(HNO3_01Mrate, HNO3_01Mrate, HNO3_01Mvol) '流速と容量を設定
    Call wait_min(separate_waittime) '溶離液通液の間、操作を待機
Next i 'Sr の溶離終了

Unload UserForm1
sample_vol = 1 + sample_vol

For i = 0 To 4 'シリンジの洗浄(硝酸)開始
    UserForm1.Label1.Caption = i + 1 & "回目の硝酸洗浄です"
    UserForm1.Show (vbModeless)
    Call valve(0, 4)
    ec.WAITmS = 300
    Call valve
    Call pump_suikomi(sample_rate, MM, sample_vol, ML)
    Call valve(1, 4)
    ec.WAITmS = 300
    Call valve
    Call pump_hakidasi(sample_rate, MM, sample_vol, ML)
Next i 'シリンジの洗浄 (硝酸)終了

For i = 0 To 3 'シリンジの洗浄(水)開始
    UserForm1.Label1.Caption = i + 1 & "回目の水洗浄です"
    Call valve(2, 0, 4)
    ec.WAITmS = 300
    Call valve
    Call pump_suikomi(sample_rate, MM, sample_vol, ML)
    Call valve(1, 4)
    ec.WAITmS = 300
    Call valve
    Call pump_hakidasi(sample_rate, MM, sample_vol, ML)
Next i 'シリンジの洗浄(水)終了

Call wash_tube 'チューブの洗浄開始
Call valve(0, 4)
ec.WAITmS = 300
Call valve
Unload UserForm1
MsgBox "分離が完了しました"

```


End Sub

Sub program_interrupt()

Dim ComPort As Byte 'ポート変数宣言
ComPort = 1 '電動バルブのポート番号を設定
ec.COMn = ComPort 'ポートを開く

ec.Ascii = "S"
ec.WAITmS = 100

ec.AsciiLine = "REM OFF"
Call valve(0, 4)

ec.WAITmS = 200
Call valve
End

End Sub

Sub separateprogram_interrupt()

ec.COMn = 1 '電動バルブのポート番号を設定
ec.AsciiLine = "0 STP"
ec.COMn = 1 '電動バルブのポート番号を設定

ec.Ascii = "S"
ec.COMn = 1 '電動バルブのポート番号を設定

ec.Ascii = "S"
Call valve(4, 0)
ec.WAITmS = 200

Call valve
ec.COMnClose = -1

End

End Sub

Function wash_tube()

MsgBox "廃液入れ以外のチューブを水に入れてください"

For i = 0 To 2

Call valve(0, 1, 4)
ec.WAITmS = 300
Call valve
Call pump_suikomi(6, MM, 3, ML)
ec.WAITmS = 500
Call valve(0, 4)
ec.WAITmS = 300
Call valve
Call pump_suikomi(6, MM, 3, ML)
ec.WAITmS = 500
Call valve(2, 4)
ec.WAITmS = 300
Call valve
Call pump_suikomi(6, MM, 3, ML)
ec.WAITmS = 500
Call valve(2, 0, 4)
ec.WAITmS = 300
Call valve
Call pump_suikomi(6, MM, 3, ML)
ec.WAITmS = 500
Call valve(1, 4)
ec.WAITmS = 300
Call valve
Call pump_hakidasi(12, MM, 12, ML)
ec.WAITmS = 100

Next i

MsgBox "チューブの先端を廃液入れに入れてください"

Call valve(2, 1, 4)

ec.WAITmS = 300

Call valve

Call pump_suikomi(20, MM, 20, ML)

Call valve(0, 1, 4)

ec.WAITmS = 300

Call valve

Call pump_hakidasi(12, MM, 5, ML)

Call valve(0, 4)

ec.WAITmS = 300

Call valve

Call pump_hakidasi(12, MM, 5, ML)

Call valve(2, 4)

ec.WAITmS = 300

Call valve

Call pump_hakidasi(12, MM, 5, ML)

Call valve(0, 2, 4)

ec.WAITmS = 300

Call valve

Call pump_hakidasi(12, MM, 5, ML)

End Function

A2-2.12 モジュール⑥ 補助用プログラム

Public Declare Sub Sleep Lib "Kernel32" (ByVal dwMilliseconds As Long)

Function wait(ByVal time As Long) '待機用プログラム、読み込む際は括弧内に変数、定数を入力

Dim i As Integer

For i = 0 To 999 '0 から 999 まで、1000 回のループ

DoEvents 'OS に処理を移す

Sleep time 'time 分だけ待ちを入れる

Next i

End Function

Function wait_min(ByVal time_min As Long)

Dim i As Integer

For i = 0 To 999

If i Mod 10 = 0 Then

UserForm1.Label2.Caption = "待機中" & i / 10 & "%"

End If

DoEvents

Sleep time_min

Next i

UserForm1.Label2.Caption = ""

End Function

Function Record(ByVal Pu As Integer) 'ログの記録用プロシージャ

Worksheets("Sheet3").Activate

Dim i As Integer

Dim Rectext(9)

i = 0

Do While Cells(i + 1, 1) <> ""

```

    DoEvents
    i = 1 + i
Loop
Cells(i + 1, 1) = Now

If Pu = 0 Then
    Worksheets("Sheet1").Activate

    For j = 0 To 4
        Rectext(j) = Cells(j + 3, 4)
    Next j

    Worksheets("Sheet3").Activate

    For j = 0 To 4
        Cells(i + 1, j + 2) = Rectext(j)
    Next j

    ElseIf Pu = 1 Then

        Worksheets("Sheet2").Activate

        For j = 0 To 9
            Rectext(j) = Cells(j + 3, 4)
        Next j

        Worksheets("Sheet3").Activate
        For j = 0 To 9
            Cells(i + 1, j + 6) = Rectext(j)
        Next j
    End If
End Function

Function unit_change()
    Call main
    ec.COMn = 1
    Call dosimat_separate(60, 60, 20)
End Function

Sub pump_adjust()
    For i = 0 To 5
        Call pump_suikomi(3, MM, 1, ML)
        Call pump_hakidasi(3, MM, 1, ML)
    Next i
End Sub

Function main() 'COM ポートを開き通信設定を行う関数
    Call valve
    ec.COMn = 1 'シリンジポンプの通信設定、ポート番号の設定
    ec.Delimiter = ec.DELIMs.Cr 'デリミタの設定
    ec.Setting = "9600,n,8,1" '通信条件の設定
    ec.AsciiLineTimeOut = 1500 'タイムアウト設定
    ec.HandShaking = "N" 'ハントシェイクなし
    ec.COMn = 1 '3M 硝酸用電動ビュレットの通信設定、ポート番号の設定
    ec.Setting = "9600,e,7,1" '通信条件の設定
    ec.WAITmS = 500
    ec.AsciiLineTimeOut = 1500 'タイムアウト設定
    ec.WAITmS = 500
    ec.HandShaking = "X" 'ハントシェイク XOn/XOff
    ec.WAITmS = 500
    ec.Delimiter = ec.DELIMs.CrLf 'デリミタに CrLf を設定
    ec.WAITmS = 500 '待機時間 500ms

    ec.AsciiLine = "REM ON" 'リモート機能 ON
    ec.WAITmS = 4000

```

```

ec.COMn = 1 '0.01M 硝酸用電動ビュレット通信設定、ポート番号の設定
ec.Setting = "9600,e,7,1" '通信条件の設定
ec.WAITmS = 500
ec.AsciiLineTimeOut = 1500 'タイムアウト設定
ec.WAITmS = 500
ec.HandShaking = "X" 'ハンドシェイク XOn/XOff
ec.WAITmS = 500
ec.Delimiter = ec.DELIMs.CrLf 'デリミタに CrLf を設定
ec.WAITmS = 500 '待機時間 500ms

ec.AsciiLine = "REM ON" 'リモート機能 ON
ec.WAITmS = 4000

ec.COMn = 1 'フラクションコレクター通信設定、ポート番号の設定
ec.Setting = "9600,n,8,1" '通信条件の設定
ec.AsciiLineTimeOut = 1500 'タイムアウト設定
ec.HandShaking = "N" 'ハンドシェイクなし
ec.Delimiter = ec.DELIMs.CrLf
ec.OutBuffer = 100& * 1024&
End Function

```

国際単位系（SI）

表 1. SI 基本単位

基本量	SI 基本単位	
	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質량	モル	mol
光度	カンデラ	cd

表 2. 基本単位を用いて表されるSI組立単位の例

組立量	SI 組立単位	
	名称	記号
面積	平方メートル	m ²
体積	立方メートル	m ³
速度	メートル毎秒	m/s
加速度	メートル毎秒毎秒	m/s ²
波数	毎メートル	m ⁻¹
密度, 質量密度	キログラム毎立方メートル	kg/m ³
面積密度	キログラム毎平方メートル	kg/m ²
比体積	立方メートル毎キログラム	m ³ /kg
電流密度	アンペア毎平方メートル	A/m ²
磁界の強さ	アンペア毎メートル	A/m
量濃度 ^(a) , 濃度	モル毎立方メートル	mol/m ³
質量濃度	キログラム毎立方メートル	kg/m ³
輝度	カンデラ毎平方メートル	cd/m ²
屈折率 ^(b)	(数字の) 1	1
比透磁率 ^(b)	(数字の) 1	1

(a) 量濃度 (amount concentration) は臨床化学の分野では物質濃度 (substance concentration) ともよばれる。

(b) これらは無次元量あるいは次元 1 をもつ量であるが、そのことを表す単位記号である数字の 1 は通常は表記しない。

表 3. 固有の名称と記号で表されるSI組立単位

組立量	SI 組立単位			
	名称	記号	他のSI単位による表し方	SI基本単位による表し方
平面角	ラジアン ^(b)	rad	1 ^(b)	m/m
立体角	ステラジアン ^(b)	sr ^(c)	1 ^(b)	m ² /m ²
周波数	ヘルツ ^(d)	Hz		s ⁻¹
力	ニュートン	N		m kg s ⁻²
圧力, 応力	パスカル	Pa	N/m ²	m ⁻¹ kg s ⁻²
エネルギー, 仕事, 熱量	ジュール	J	N m	m ² kg s ⁻²
仕事率, 工率, 放射束	ワット	W	J/s	m ² kg s ⁻³
電荷, 電気量	クーロン	C		s A
電位差 (電圧), 起電力	ボルト	V	W/A	m ² kg s ⁻³ A ⁻¹
静電容量	ファラド	F	C/V	m ⁻² kg ⁻¹ s ⁴ A ²
電気抵抗	オーム	Ω	V/A	m ² kg s ⁻³ A ⁻²
コンダクタンス	ジーメンズ	S	A/V	m ⁻² kg ⁻¹ s ³ A ²
磁束	ウェーバ	Wb	Vs	m ² kg s ⁻² A ⁻¹
磁束密度	テスラ	T	Wb/m ²	kg s ⁻² A ⁻¹
インダクタンス	ヘンリー	H	Wb/A	m ² kg s ⁻² A ⁻²
セルシウス温度	セルシウス度 ^(e)	°C		K
光束度	ルーメン	lm	cd sr ^(c)	cd
照射度	ルクス	lx	lm/m ²	m ⁻² cd
放射性核種の放射能 ^(f)	ベクレル ^(d)	Bq		s ⁻¹
吸収線量, 比エネルギー分与, カーマ	グレイ	Gy	J/kg	m ² s ⁻²
線量当量, 周辺線量当量, 方向性線量当量, 個人線量当量	シーベルト ^(g)	Sv	J/kg	m ² s ⁻²
酸素活性化	カタール	kat		s ⁻¹ mol

(a) SI接頭語は固有の名称と記号を持つ組立単位と組み合わせても使用できる。しかし接頭語を付した単位はもはやコヒーレントではない。

(b) ラジアンとステラジアンは数字の 1 に対する単位の特別な名称で、量についての情報をつたえるために使われる。実際には、使用する時には記号rad及びsrが用いられるが、習慣として組立単位としての記号である数字の 1 は明示されない。

(c) 測光学ではステラジアンという名称と記号srを単位の表し方の中に、そのまま維持している。

(d) ヘルツは周期現象についてののみ、ベクレルは放射性核種の統計的過程についてののみ使用される。

(e) セルシウス度はケルビンの特別な名称で、セルシウス温度を表すために使用される。セルシウス度とケルビンの単位の大きさは同一である。したがって、温度差や温度間隔を表す数値はどちらの単位で表しても同じである。

(f) 放射性核種の放射能 (activity referred to a radionuclide) は、しばしば誤った用語で"radioactivity"と記される。

(g) 単位シーベルト (PV, 2002, 70, 205) についてはCIPM勧告2 (CI-2002) を参照。

表 4. 単位の中に固有の名称と記号を含むSI組立単位の例

組立量	SI 組立単位		
	名称	記号	SI 基本単位による表し方
粘着力のモーメント	パスカル秒	Pa s	m ⁻¹ kg s ⁻¹
表面張力	ニュートンメートル	N m	m ² kg s ⁻²
角速度	ニュートン毎メートル	N/m	kg s ⁻²
角加速度	ラジアン毎秒	rad/s	m m ⁻¹ s ⁻¹ =s ⁻¹
熱流密度, 放射照度	ラジアン毎秒毎秒	rad/s ²	m m ⁻¹ s ⁻² =s ⁻²
熱容量, エントロピー	ワット毎平方メートル	W/m ²	kg s ⁻³
比熱容量, 比エントロピー	ジュール毎ケルビン	J/K	m ² kg s ⁻² K ⁻¹
比エネルギー	ジュール毎キログラム毎ケルビン	J/(kg K)	m ² s ⁻² K ⁻¹
熱伝導率	ジュール毎キログラム	J/kg	m ² s ⁻²
体積エネルギー	ワット毎メートル毎ケルビン	W/(m K)	m kg s ⁻³ K ⁻¹
電界の強さ	ジュール毎立方メートル	J/m ³	m ⁻¹ kg s ⁻²
電荷密度	ジュール毎平方メートル	V/m	m kg s ⁻³ A ⁻¹
表面電荷密度	クーロン毎立方メートル	C/m ³	m ⁻³ s A
電束密度, 電気変位	クーロン毎平方メートル	C/m ²	m ⁻² s A
誘電率	クーロン毎平方メートル	C/m ²	m ² s A
透磁率	ファラド毎メートル	F/m	m ³ kg ⁻¹ s ⁴ A ²
モルエネルギー	ヘンリー毎メートル	H/m	m kg s ⁻² A ⁻²
モルエントロピー, モル熱容量	ジュール毎モル	J/mol	m ² kg s ⁻² mol ⁻¹
照射線量 (X線及びγ線)	ジュール毎モル毎ケルビン	J/(mol K)	m ² kg s ⁻² K ⁻¹ mol ⁻¹
吸収線量率	クーロン毎キログラム	C/kg	kg ⁻¹ s A
放射線強度	グレイ毎秒	Gy/s	m ² s ⁻³
放射輝度	ワット毎ステラジアン	W/sr	m ⁴ m ⁻² kg s ⁻³ =m ² kg s ⁻³
酵素活性濃度	ワット毎平方メートル毎ステラジアン	W/(m ² sr)	m ² m ⁻² kg s ⁻³ =kg s ⁻³
	カタール毎立方メートル	kat/m ³	m ⁻³ s ⁻¹ mol

表 5. SI 接頭語

乗数	名称	記号	乗数	名称	記号
10 ²⁴	ヨタ	Y	10 ⁻¹	デシ	d
10 ²¹	ゼタ	Z	10 ⁻²	センチ	c
10 ¹⁸	エクサ	E	10 ⁻³	ミリ	m
10 ¹⁵	ペタ	P	10 ⁻⁶	マイクロ	μ
10 ¹²	テラ	T	10 ⁻⁹	ナノ	n
10 ⁹	ギガ	G	10 ⁻¹²	ピコ	p
10 ⁶	メガ	M	10 ⁻¹⁵	フェムト	f
10 ³	キロ	k	10 ⁻¹⁸	アト	a
10 ²	ヘクト	h	10 ⁻²¹	ゼプト	z
10 ¹	デカ	da	10 ⁻²⁴	ヨクト	y

表 6. SIに属さないが、SIと併用される単位

名称	記号	SI 単位による値
分	min	1 min=60 s
時	h	1 h=60 min=3600 s
日	d	1 d=24 h=86 400 s
度	°	1°=(π/180) rad
分	′	1′=(1/60)°=(π/10 800) rad
秒	″	1″=(1/60)′=(π/648 000) rad
ヘクタール	ha	1 ha=1 hm ² =10 ⁴ m ²
リットル	L, l	1 L=1 l=1 dm ³ =10 ³ cm ³ =10 ⁻³ m ³
トン	t	1 t=10 ³ kg

表 7. SIに属さないが、SIと併用される単位で、SI単位で表される数値が実験的に得られるもの

名称	記号	SI 単位で表される数値
電子ボルト	eV	1 eV=1.602 176 53(14)×10 ⁻¹⁹ J
ダルトン	Da	1 Da=1.660 538 86(28)×10 ⁻²⁷ kg
統一原子質量単位	u	1 u=1 Da
天文単位	ua	1 ua=1.495 978 706 91(6)×10 ¹¹ m

表 8. SIに属さないが、SIと併用されるその他の単位

名称	記号	SI 単位で表される数値
バール	bar	1 bar=0.1 MPa=100 kPa=10 ⁵ Pa
水銀柱ミリメートル	mmHg	1 mmHg=133.322 Pa
オングストローム	Å	1 Å=0.1 nm=100 pm=10 ⁻¹⁰ m
海里	M	1 M=1852 m
バイン	b	1 b=100 fm ² =(10 ¹² cm) ² =10 ⁻²⁸ m ²
ノット	kn	1 kn=(1852/3600) m/s
ネーパ	Np	SI単位との数値的な関係は、 対数量の定義に依存。
ベレル	B	
デシベル	dB	

表 9. 固有の名称をもつCGS組立単位

名称	記号	SI 単位で表される数値
エル	erg	1 erg=10 ⁻⁷ J
ダイン	dyn	1 dyn=10 ⁻⁵ N
ポアズ	P	1 P=1 dyn s cm ⁻² =0.1 Pa s
ストークス	St	1 St=1 cm ² s ⁻¹ =10 ⁻⁴ m ² s ⁻¹
スチルブ	sb	1 sb=1 cd cm ⁻² =10 ⁴ cd m ⁻²
フオット	ph	1 ph=1 cd sr cm ⁻² =10 ⁴ lx
ガリ	Gal	1 Gal=1 cm s ⁻² =10 ⁻² ms ⁻²
マクスウェル	Mx	1 Mx=1 G cm ² =10 ⁻⁸ Wb
ガウス	G	1 G=1 Mx cm ⁻² =10 ⁻⁴ T
エルステッド ^(a)	Oe	1 Oe _e (10 ³ /4 π) A m ⁻¹

(a) 3 元素のCGS単位系とSIでは直接比較できないため、等号「 \triangle 」は対応関係を示すものである。

表 10. SIに属さないその他の単位の例

名称	記号	SI 単位で表される数値
キュリー	Ci	1 Ci=3.7×10 ¹⁰ Bq
レントゲン	R	1 R=2.58×10 ⁻⁴ C/kg
ラド	rad	1 rad=1 cGy=10 ⁻² Gy
レム	rem	1 rem=1 cSv=10 ⁻² Sv
ガンマ	γ	1 γ=1 nT=10 ⁻⁹ T
フェルミ	f	1 フェルミ=1 fm=10 ⁻¹⁵ m
メートル系カラット		1 メートル系カラット=0.2 g=2×10 ⁻⁴ kg
トル	Torr	1 Torr=(101 325/760) Pa
標準大気圧	atm	1 atm=101 325 Pa
カロリ	cal	1 cal=4.1858 J (「15°C」カロリ), 4.1868 J (「IT」カロリ), 4.184 J (「熱化学」カロリ)
マイクロン	μ	1 μ=1 μm=10 ⁻⁶ m

