

電界放射実験装置の Excel VBA と GPIB による制御

川久保 貴史* 杉田 健真**

Control of Field Emission Experiment Equipment with Excel VBA and GPIB

Takashi KAWAKUBO and Kenshin SUGITA

概要

我々の研究室では、電界放射電子源の研究を行っている。電子源の特性を測定するための電界放射実験装置は、過去に研究室の学生が、Visual Basic を用いて測定装置を制御し自動で測定を行うことができるシステムを構築した。しかし、当時開発に用いた Visual Basic 6.0 は開発環境としてのサポートが終了し、現在も使い続けることにはリスクがある。そのため、新たに Excel VBA を用いた自動測定システムを構築し、その動作を検証したので報告する。

キーワード：電界放射， GPIB， Excel VBA， VISA

1. まえがき

我々の研究室では、電界放射電子源の研究を行っている。研究室では実際に作製した電子源試料を超高真空装置内に設置し、電子放射を行うことでその特性を測定する実験を行っているが、電子放射特性は時間経過とともに変化するため迅速な測定が求められ、さらに、得られた測定データの処理も煩雑である。そのため、過去には研究室の学生が、Visual Basic を用いて測定装置を制御し自動で電子放射測定を行うシステムを構築したり。しかし、当時開発に用いた Visual Basic 6.0 は開発環境としてのサポートが終了し、使い続けることにはリスクがある。そのため、Visual Basic 以外で新たな電子放射測定システムを開発する必要が生じてきた。本研究では、Excel VBA を用いて新たなシステムを構築し、その動作を検証したので報告する。

2. 電子放射測定システム

図 1 に電界放射実験装置の全体像を示す。ステンレス製のチャンパー内はロータリーポンプ、ターボ分子ポンプ、イオンポンプによる排気と 11 時間のベーキングによって 10^{-7} Pa オーダの到達圧が得られる。チャン

パー内には、研究室で作製した電子源試料を設置し、それに対向するように蛍光板（陽極）を設置する。

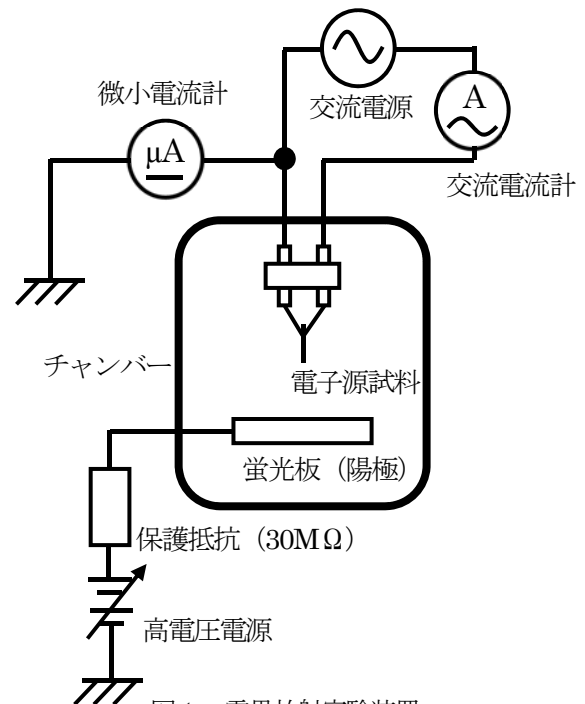


図 1 電界放射実験装置

電子源試料と蛍光板は、チャンパーに取り付けられた導入端子によってチャンパー外の回路と接続するこ

* 香川高等専門学校 通信ネットワーク工学科

** 香川高等専門学校専攻科 電子情報通信工学専攻

とができる。電子源試料には、試料を通電加熱するための交流電源とその電流を測定する電流計（交流）を接続する。また、電子源試料と蛍光板間に高電圧を印加し電界放射させるための高電圧電源（直流）とその放射電流を測定する微小電流計（直流 μA 計）を接続する。保護抵抗（ $30\text{M}\Omega$ ）は高電圧印加時にチャンバー内で放電現象が起こった時に大電流が流れるのを制限するために接続している。本研究で制御する実験装置は、電界放射のための高電圧電源と放射電流を測定する微小電流計である。

2.1. GPIB について

GPIB（General Purpose Interface Bus）は、PC と計測器をつなぐ国際標準規格のインターフェイスである。元はヒューレット・パッカード社の社内規格 HPIB として開発されたが、後に IEEE（Institute of Electrical and Electronic Engineers）委員会によって承認され、国際標準規格 GPIB（または IEEE 488 バスとも呼ばれる）となった²⁾。現在、GPIB インターフェイスを標準で搭載した計測器は多く、PC を用いた計測器の制御に GPIB は広く用いられている²⁾。

本研究室で用いている高電圧電源（松定プレジジョン社製）と微小電流計（ADVANTEST 社製）にも GPIB 接続できる端子が付いており、過去に学生が構築した電子放射測定システムは、各機器の GPIB 端子からケーブルで PC と接続し制御を行っていたり、それらを参考に、今回も GPIB による制御を試みた。

GPIB は複数の計測器をデジチェーン方式で接続することができるが、その際に、各計測器に固有アドレスを設定する必要がある。今回は、高電圧電源の固有アドレスを 1 番に、微小電流計の固有アドレスを 3 番に設定した³⁾⁴⁾。

GPIB を PC に接続するために GPIB-USB 変換ケーブルを用いた。これまでは、CONTEC 社製のケーブル（GP-IB(USB)FL）を用いていたが、今回新たに構築したシステムでは、PC と計測器間の通信がうまくいかず、図 2 に示すアジレント社製のケーブル（82357B USB/GPIB）へ変更した。

2.2. VISA について

本研究では、計測器と PC 間の通信ドライバに VISA ライブラリを用いた。VISA（Virtual Instrument Software Architecture）は計測器を制御するためのライブラリで、世界の共通規格として IVI（Interchangeable Virtual Instruments）協会が制定されており、現在、多くの計測器で用いることができる⁵⁾。また VISA はいくつかの計

測器メーカーから供給されており、各社のウェブページから無料、あるいは条件付きでダウンロードして使うことができるようになっている⁵⁾。

今回は、キーサイト・テクノロジー社製（アジレント社の電子計測事業がキーサイト・テクノロジー社）の VISA を用いた⁵⁾。ウェブページから Keysight IO Libraries Suite 一式をダウンロードし PC へインストールした。図 3 に IO Libraries Suite のコントロール画面を示す。PC と計測器の接続状況が表示されている。この場合では、2 台の計測器（高電圧電源と微小電流計）が接続されている。また、この VISA は GPIB 以外にも RS-232C や USB 接続にも対応しており、他の計測器のインターフェイスも用いることができる。



図2 アジレント社製 GPIB-USB 変換ケーブル

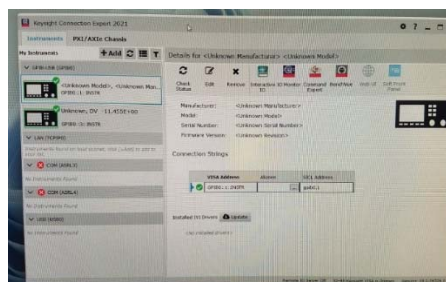


図3 コントロール画面

2.3. 高電圧電源について

電子源には複数の電子放射モードがあるが、我々の研究室では電界放射モードを主に扱っている。電界放射は、電子源試料の電子放射面に高電界を印加し電子放射をさせるモードで、高電界を生じさせるために、数 kV の高電圧が必要となる。本研究では、松定プレジジョン製のプログラマブル高電圧直流電源（HAR 正極性タイプ）に、GPIB 用光接続用アダプタ（GP-32）をつなぎ GPIB で使用した。

2.4. 微小電流計について

電界放射実験では、電子源試料からの電子放射を電流として微小電流計で測定している。本研究では、ADVANTEST 社製のデジタル・エレクトロメータ R8240 を直流電流測定（ μA ）モードで用いた。

2.5. Excel VBA について

VBA (Visual Basic for Applications) は Microsoft Office で利用できるプログラミング言語である。Office 製品に標準搭載されているので、Microsoft Office が既にインストールされている研究室の PC では、新たな開発環境を導入しなくても使用することができる。

本研究室には、プログラミングが得意でないメンバーが多く、新たな開発環境を導入するには敷居が高い。また、本研究では、測定後のデータを表計算ソフトで処理する必要もある。そのため、今回は研究室の PC に最初から環境が備わっている VBA を用い、表計算ソフト Excel 上で動作するプログラムを作成することにした。

2.6. 測定後のデータ整理について

本実験では、電子源試料に印加した電圧 V とその時の放射電流 I のデータ (I - V 特性) を記録する。さらに、その結果を電界放射の理論 Fowler-Nordheim の式に基づいて、表計算ソフト Excel で計算を行い、F-N プロットを作成する。

Fowler-Nordheim の式を(1)式に示す。式中の J は放射電流密度、 E は電界強度、 ϕ は仕事関数である。また、 t や v は補正係数であり、大きく変化する値ではないため 1 として近似し計算されることがある。

$$J = \left\{ \frac{[1.54 \times 10^{-6} E^2]}{[\phi t^2(v)]} \right\} \exp \left\{ -6.83 \times \frac{10^7 \phi^2}{[Ev(v)]} \right\} \quad (1)$$

今回のシステムは、測定した I - V 特性を、電界強度 E と放射電流密度 J の関係に変換して(1)式へ代入・式変形を行い、電子源試料の仕事関数 ϕ を求められるようにした。仕事関数 ϕ は、金属中のフェルミ準位にある電子を真空中へ取出すのに必要なエネルギーとされ、仕事関数が低いほど電子放射しやすい電子源試料であることがわかる。

3. 作成した制御プログラムと動作

Excel VBA のプログラムは、Visual Basic Editor を起動し、ツールボックスからユーザーフォームを挿入した後、その上にコントロール (ボタン) を配置したり、取り込んだ値を格納する Excel のセルを指定したりすることで操作画面が作成できる。さらに、Visual Basic Editor のツール設定から参照設定を選び、通信ドライバとして VISA ライブラリを設定することで、作成したプログラムで計測器を制御できるようになる。

GPIB で接続した計測器は、PC 側から ASCII 文字列のコマンドを送ることで動作を指定し、また、計測器

側から ASCII 文字列として測定値や測定状態を PC 側に返すやり取りを行う。そのため、VBA で作成した操作画面のコントロールボタンや Excel セルには、そのやり取りを指示するプログラムを書き込んでいくことになる。PC と計測器間でやり取りするコマンドは、計測器の取扱説明書を参考にした³⁾⁴⁾。

```
Sub denryu_20micro()
    Dim RM As New VisaComLib.ResourceManager
    Dim DMM As New VisaComLib.FormattedIO488
    Set DMM.IO = RM.Open("GPIB0::3::INSTR")

    DMM.WriteString "F2 R7"
    DMM.IO.Close

    Set DMM = Nothing
    Set RM = Nothing
End Sub
Sub tyousei()
    Dim RM As New VisaComLib.ResourceManager
    Dim DMM As New VisaComLib.FormattedIO488
    Set DMM.IO = RM.Open("GPIB0::3::INSTR")

    DMM.WriteString "MD1"
    Application.Wait Now + TimeValue("00:00:04")
    DMM.WriteString "MD0"
    DMM.IO.Close

    Set DMM = Nothing
    Set RM = Nothing
End Sub
```

図4 レンジ変更とゼロチェックのプログラム

図4は、Excel VBA で作成した微小電流計のレンジ変更とゼロチェックのプログラム例である。Sub から End Sub までが1つのボタンに書き込まれたプログラムである。

Sub denryu_20micro()について解説する。最初の Dim RM~, および、Dim DMM~の部分で、今回用いたキーワード・テクノロジー社製の VISA ライブラリを呼び出し、Set DMM.IO=RM Open で、計測器のアドレスを指定し通信を開始している。今回は微小電流計に指定した固有アドレス 3 番が設定されている。次の DMM.WriteString で、計測器へ文字列を送っており、「F2 R7」という文字列で、微小電流計に「直流電流測定、20 μ A レンジ」での測定を指定している。DMM.IO.Close で通信を終了し、Set DMM と Set RM をともに Nothing とすることで、呼び出していた VISA を開放している。

Sub tyousei()についても、基本的に似た構造となっており、「MD1」の文字列は、使用した微小電流計において、「ゼロチェック ON」にすることを意味し、「MD0」は「ゼロチェック OFF」にすることを意味している。その間の Application.Wait 部分において、4秒待つように指示しているため、ゼロチェックを ON にして4秒待った後 OFF にすることになる。

```

Sub sokutei_3_4() '測定 3~4kV
    Dim RM As New VisaComLib.ResourceManager
    Dim DMM As New VisaComLib.FormattedIO488
    Dim AMM As New VisaComLib.FormattedIO488
    Dim i As Long
    Set DMM.IO = RM.Open("GPIB0::1::INSTR")
    Set AMM.IO = RM.Open("GPIB0::3::INSTR")
    Dim c3 As Integer
    Dim c4 As Integer
    Dim v As Variant
    c4 = 1
    c3 = 1

    Worksheets("測定").Range("B3").Value = "3-4kV"

    For i = 3 To 13

        DMM.WriteString "#1 REN"
        DMM.WriteString "#1 ICN 100.00"
        DMM.WriteString "#1 SW1"
        Select Case i

            Case 3
                DMM.WriteString "#1 VCN 30.00"

            Case 4
                DMM.WriteString "#1 VCN 31.00"
                c4 = c4 + 1

            Case 5
                DMM.WriteString "#1 VCN 32.00"
                c4 = c4 + 1

            Case 6
                DMM.WriteString "#1 VCN 33.00"
                c4 = c4 + 1

            Case 7
                DMM.WriteString "#1 VCN 34.00"
                c4 = c4 + 1

            Case 8
                DMM.WriteString "#1 VCN 35.00"
                c4 = c4 + 1

            Case 9
                DMM.WriteString "#1 VCN 36.00"
                c4 = c4 + 1

            Case 10
                DMM.WriteString "#1 VCN 37.00"
                c4 = c4 + 1

            Case 11
                DMM.WriteString "#1 VCN 38.00"
                c4 = c4 + 1

            Case 12
                DMM.WriteString "#1 VCN 39.00"
                c4 = c4 + 1

            Case 13
                DMM.WriteString "#1 VCN 40.00"
                c4 = c4 + 1

        End Select
        DMM.WriteString "#1 VCN?"
        Cells(i, 5) = DMM.ReadString()
        For c3 = 1 To 4 'loop

            AMM.WriteString "READ?"
            Worksheets("値保管所").Cells(c3, c4) = AMM.ReadString()

            Next 'finish loop
        Next

        DMM.WriteString "#1 SW0"
        DMM.WriteString "#1 GTL"
        DMM.IO.Close
        AMM.IO.Close
    
```

```

    Dim volt(10) As Long
    Dim cc As Integer
    volt(0) = 3000
    volt(1) = 3100
    volt(2) = 3200
    volt(3) = 3300
    volt(4) = 3400
    volt(5) = 3500
    volt(6) = 3600
    volt(7) = 3700
    volt(8) = 3800
    volt(9) = 3900
    volt(10) = 4000
    For c2 = 3 To 13

        Worksheets("グラフ").Cells(c2, 3).Value = volt(cc)
        cc = cc + 1
    Next

    Dim retu As Integer
    Dim count As Integer
    Dim list(10) As String
    list(0) = "3.0kV"
    list(1) = "3.1kV"
    list(2) = "3.2kV"
    list(3) = "3.3kV"
    list(4) = "3.4kV"
    list(5) = "3.5kV"
    list(6) = "3.6kV"
    list(7) = "3.7kV"
    list(8) = "3.8kV"
    list(9) = "3.9kV"
    list(10) = "4.0kV"
    For retu = 3 To 13
        Cells(retu, 3) = list(count)
        count = count + 1
    Next

    Dim c5 As Integer
    Dim c6 As Integer
    Dim stRight As String
    c6 = 6
    b = 1
    Dim bb As Integer
    For bb = 1 To 11
        For c5 = 1 To 4
            Worksheets("値保管所").Activate
            stRight = Right(Worksheets("値保管所").Cells(c5, bb), 12)
            Worksheets("値保管所").Cells(c6, bb).Value = stRight
            c6 = c6 + 1
        Next
        c6 = 6
    Next

    Dim cc7 As Integer
    Dim c8 As Integer
    c8 = 10
    Dim a As Integer
    For a = 1 To 11
        For cc7 = 6 To 9
            v = Val(Worksheets("値保管所").Cells(cc7, a))
            Worksheets("値保管所").Cells(c8, a) = v
            c8 = c8 + 1
        Next
        c8 = 10
    Next
End Sub

```

図5 測定プログラム (3~4 kV 印加)

図5に測定部のプログラムを示す。ここでは高電圧 V の印加、放射電流値 I の測定、測定結果の整理を行っている。PCから高電圧電源へ数kVの高電圧 V を発生させるように命令を送る。すると、実験装置の電子源試料と対向する蛍光板の間へ高電圧が印加され、電子源試料で電界放射現象が発生し電子放射が起こる。この放射電流を微小電流計で測定した値 I がPCへ返ってくる。図5のプログラムでは、印加電圧 V は3.0 kVから100 V毎に上昇させて4.0 kVまで11点の測定点を取るようになっている。

各高電圧 V を印加した時に微小電流計で測定された電流値 I をPCへ取り込み、Excelシートに記録されるようにした。測定は、設定した各高電圧につき、1つの電圧が印加されている間に4回の電流測定を行い、その平均値を、印加電圧に対する電流値として記録している。(例えば、印加電圧 $V=3.3$ kVの時に、放射電流 I が $0.11 \mu\text{A}$, $0.11 \mu\text{A}$, $0.13 \mu\text{A}$, $0.09 \mu\text{A}$ の4回が得られた場合、4回の電流値の平均をとって $0.11 \mu\text{A}$ が、印加電圧 $V=3.3$ kVに対する電流値 $I=0.11 \mu\text{A}$ となる。) Excelに出力する際に、測定された値が文字列として返されるため、一度文字列の数字部分を切り取り、その数字部分の文字列を数値に変換するという処理を行っている。

図6がExcel上のコントロール画面である。図5で示した測定プログラムは印加電圧3-4kVの範囲で測定を行うものであったが、電子源試料の放射電流量に応じて、印加電圧範囲を各ボタンで変更できるようにした。図7は本システムの測定結果から描かれたF-Nプロット例である。横軸は印加電圧 V の逆数、縦軸は放射電流 I を印加電圧 V の二乗で割ってLogで示した値である。電界放射現象が起こっているときの I - V 特性からF-Nプロットを作成すると、その結果はグラフにおいて、右下がりの直線状に並び、その直線の傾きから電子源試料の仕事関数 ϕ を求めることができる。

4. 結論

本研究では、Excel VBAを用いて電子放射測定を行うシステムを構築し、その動作を検証した。Excel VBAで高電圧電源と微小電流計を制御することで測定の自動化を試みた。高電圧電源と微小電流計をPCにGPIBで接続し制御することができた。研究の成果として、手動で行うと煩雑で時間のかかる実験手順や結果の整理を自動化することで大幅な効率化をすることができた。今後、本システムの改善、改良を行い、さらに使いやすいシステムを目指したいと考えている。



図6 コントロール画面

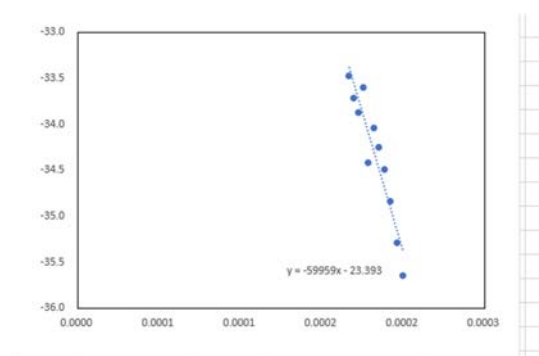


図7 F-Nプロット例

参考文献

- 1) 佐々木健太：Visual Basicによる計測器の制御，香川高等専門学校卒業論文（2012）
- 2) CONTEC：GPIB 通信の基礎知識と用語集，<https://www.contec.com/jp/support/basic-knowledge/daq-control/gpib-communication/>（2023-3-25 閲覧）
- 3) ADVANTEST：R8240 デジタル・エレクトロメータ取扱説明書，6章 pp.1-25（2003年購入品）
- 4) 松定プレジジョン：GP シリーズ取扱説明書，pp.14-15，pp.24-47（2012年購入品）
- 5) Keysight：自動計測における制御ライブラリの選択（VISA、VISA-COM），<https://www.keysight.com/jp/ja/lib/resources/programming-guides/resource-2304677.html>（2023-3-25 閲覧）