

実用的な自動計測 マルチメータ 編

キーサイト テクノロジー 株式会社

2020.04.24

石井 幹



実用的な自動計測 マルチメータ 編

目次

- デジタルマルチメータ 概要
- デジタルマルチメータ動作原理
 - DC 電圧測定
 - DC 電流測定
 - 抵抗測定
- デジタルマルチメータ 応用例
 - 高速サンプリングかつ長時間測定を実現するアプリケーション
 - DMM に必要な機能
 - アプリケーションに必要な機能
 - サンプルプログラムの紹介
 - サンプルプログラムの実行例
- デジタルマルチメータ Q&A



デジタルマルチメータ 概要

デジタルマルチメータ概要

- デジタルマルチメータの測定ファンクション
 - DC 電圧/電流、抵抗測定、AC 電圧/電流、温度、周波数、容量、...
- デジタルマルチメータの性能
 - 表示桁数
 - 確度仕様
- デジタルマルチメータの測定補助機能
 - PCから制御するためのインターフェース
 - 演算機能
 - グラフ表示機能
 - 測定値を保存するメモリ
 - 内部タイマによる一定時間ごとの測定
 - 測定中にコマンドの処理を行う機能

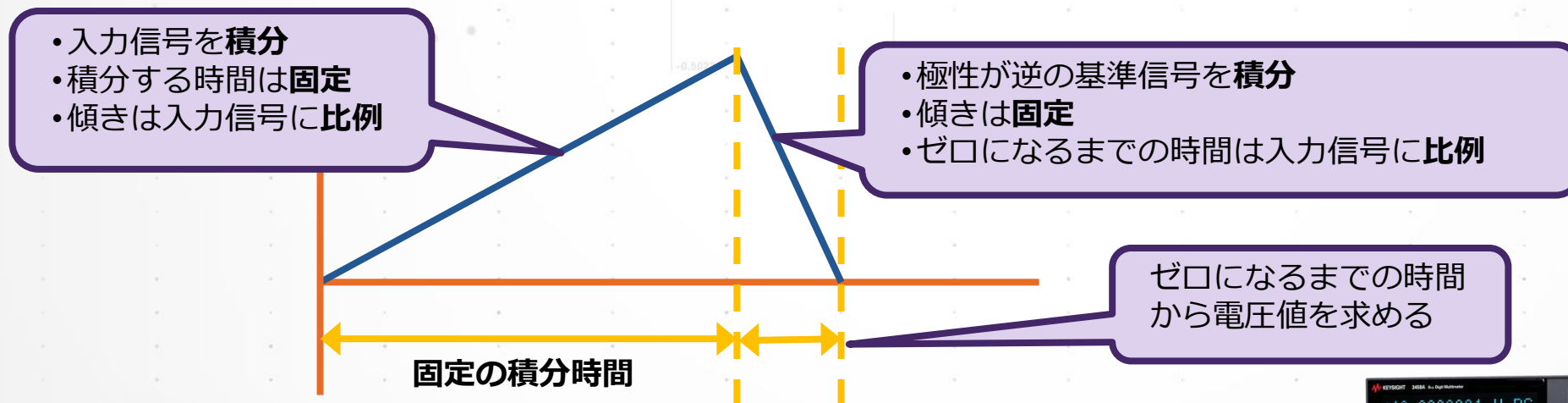


デジタルマルチメータ 動作原理

DCV

DC 電圧測定

- デジタルマルチメータのAD変換方式
 - 積分型 6・1/2桁から8・1/2桁のマルチメータで採用
 - シグマデルタ型 4・1/2桁、5・1/2桁のマルチメータで採用
- 積分型ADコンバータの原理
 - 入力信号を一定の時間（積分時間）だけ積分器で積分する
 - 基準信号を逆の極性で積分し、積分器の出力がゼロになるまでの時間を測定する



デジタルマルチメータ 動作原理

DC 電圧測定

- DC電圧測定の設定
 - 積分時間
 - PLC (Power Line Cycle)、NPLC (Number of Power Line Cycle) で指定
 - NPLC は 1 以下も指定可能
 - 1以上を指定することで、電源周波数と同期したノイズがキャンセル可能
 - 積分時間を長くするとノイズの少ない測定が可能。ただし、測定に時間がかかる
 - 自動ゼロ (オート・ゼロ)
 - 入力回路に存在するオフセット電圧をキャンセルする機能
 - ON にすると、信号に加え、内部オフセット電圧も測定するため、測定時間がかかる
 - 分解能の指定 (桁数の指定)
 - 指定された桁数を実現するために必要な積分時間が設定される
- ポイント
 - 信号の入力端子はフローティングです。絶縁は300Vから1000V程度
 - 入力端子の入力抵抗は 10M Ω 。レンジによって、10G Ω の選択が可能

DCV

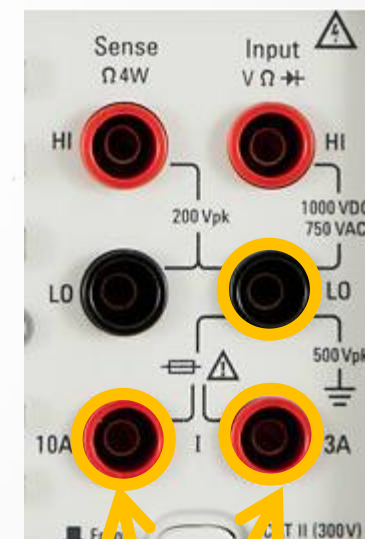
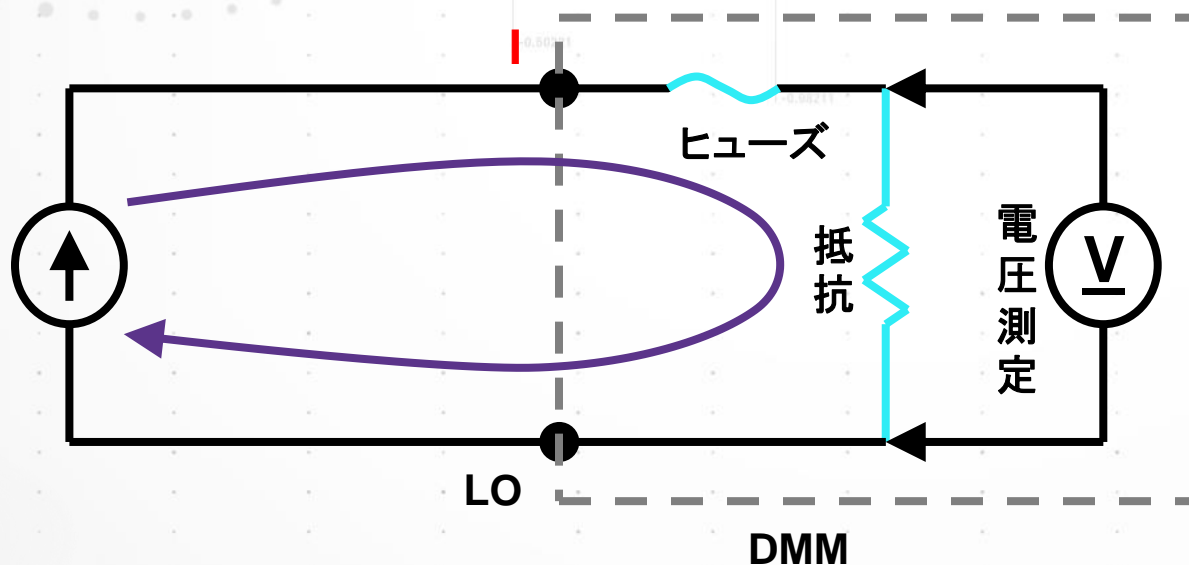


デジタルマルチメータ 動作原理

DCI

DC 電流測定

- DC電流測定の原理
 - 電流経路に電流検出用の抵抗を挿入し、抵抗に発生する電圧を測定する
- DC 電流測定の設定
 - DC 電圧測定と同様、積分時間、自動ゼロ、分解能
- ポイント
 - 電流経路に抵抗、ヒューズなどが挿入されるため、経路に影響が発生する可能性がある
 - ヒューズ切れは、マルチメータでは検出できない。（ゼロに近い電流値が測定される）

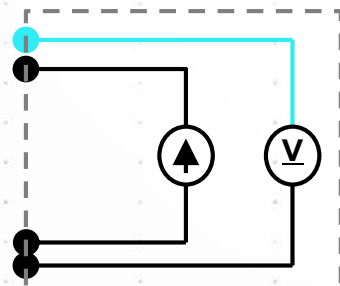
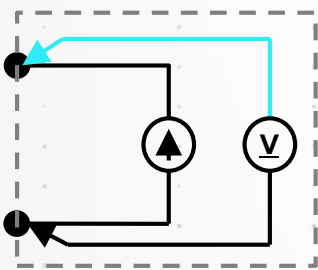


入力電流に合わせて選択

デジタルマルチメータ 動作原理

2線式/4線式 抵抗測定

- 2線式/4線式抵抗測定の設定
 - DC 電圧測定と同様 積分時間、自動ゼロ、分解能
 - オフセット補正
 - 通常の測定に加えて定電流をOFFにして経路のオフセット電圧を測定
 - オフセット電圧を抵抗測定の電圧から減算することで経路のオフセットをキャンセル
- ポイント
 - 入力端子が解放されている場合
 - 抵抗測定のための定電流源は、電圧を最大にしても電流が流せない
 - 2線式抵抗測定では、最大の電圧値が測定されるために、オーバーロードが検出される
 - 4線式抵抗測定では、解放端の電圧が測定されるために、オーバーロードは検出されない



RES/
FRES



デジタルマルチメータ応用例

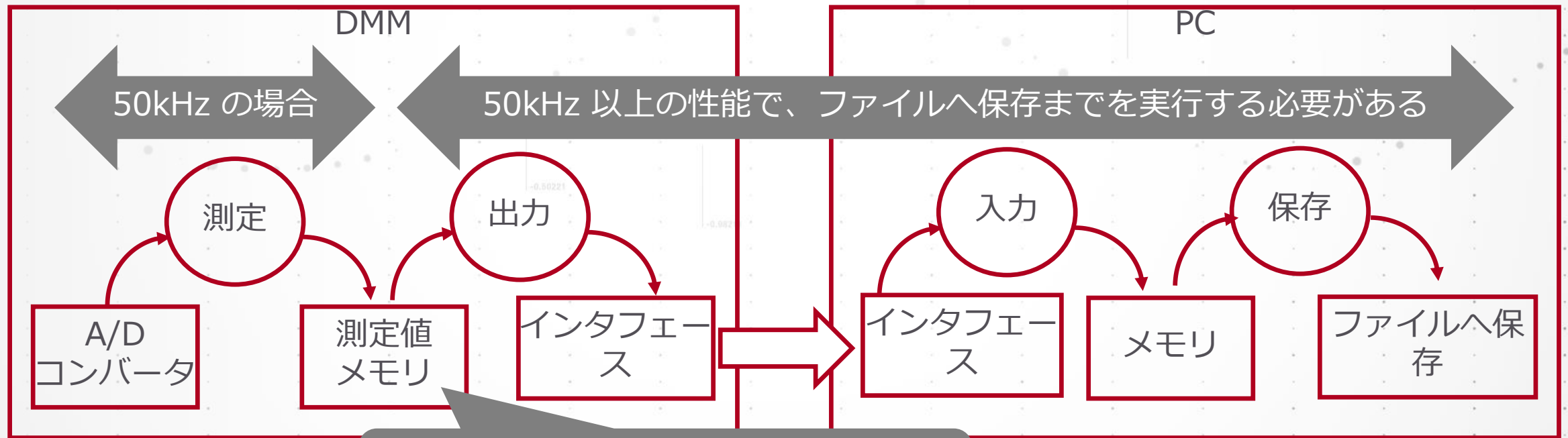
高速サンプリングかつ長時間測定を実現するアプリケーション

- 50kHz で 30分間測定を行なうなど、高速サンプリングかつ長時間測定を実現するには、DMM とアプリケーションプログラムの両方に、特定の機能、性能が必要です。
- DMM に必要な機能
 - 内蔵タイマにより一定時間間隔で測定を実施し、測定値をメモリに保存する機能
 - 測定中にメモリから測定値を PC に転送し、転送済みの測定値をメモリから削除する機能
 - 測定速度より高速に測定値を転送するインタフェース
- 34465A 6・1/2桁、34470A 7・1/2桁 デジタルマルチメータ
 - オプション DIG： 内蔵タイマによる一定時間間隔での測定。最高 50 kHzで測定可能。
 - オプション MEM: 内蔵メモリを 50kSample から、2MSample に増加
 - LAN（HiSLIP または ソケット）で PC と接続することで、50 kSa/秒以上の性能で測定値の転送が可能

デジタルマルチメータ応用例

高速サンプリングかつ長時間測定を実現するアプリケーション

- アプリケーションプログラムに必要な機能
 - サンプリング速度以上の性能で、測定値の転送と保存を行なう機能
 - DMM のメモリを監視し、オーバーフローを検出する機能



データ転送が遅れると、
測定値メモリがオーバーフロー？

デジタルマルチメータ応用例

高速サンプリングかつ長時間測定を実現するアプリケーション

• サンプルプログラム

DMMから一度に転送するサンプル数

測定開始ボタンで転送する回数

実際に測定される回数は、
「サンプル数/転送」
×
「転送回数/測定開始ボタン」
です。

VISA アドレス: TCPIP0:169.254.4.70:hislip0:INSTR

サンプル数 / 転送: 10000 測定開始

転送回数 / 測定開始ボタン: 100

サンプリング周期 (秒): 20E-6 測定停止

データログファイル名: DmmDataLog.txt 表示をクリア

DMM Memory

20000
15000
10000
5000
0

0 20 40 60 80 100

転送済みの回数: 100

DMM メモリの最大使用量: 15792

DMM オープン成功
*IDN?: Keysight Technologies,34470A,MY57701384,A.03.00-02.40-03.00-00.52-04-01
SYST:ERR?: +0, "No error"
DMM 測定開始 成功
測定完了
SYST:ERR?: +0, "No error"

DMM の VISA アドレス

測定パラメータを設定し、測定を開始

測定を中断

DMMのメモリ使用量のグラフ
時間経過とともに増加している場合、
メモリがオーバーフローする可能性がある

現時点で測定が完了した回数

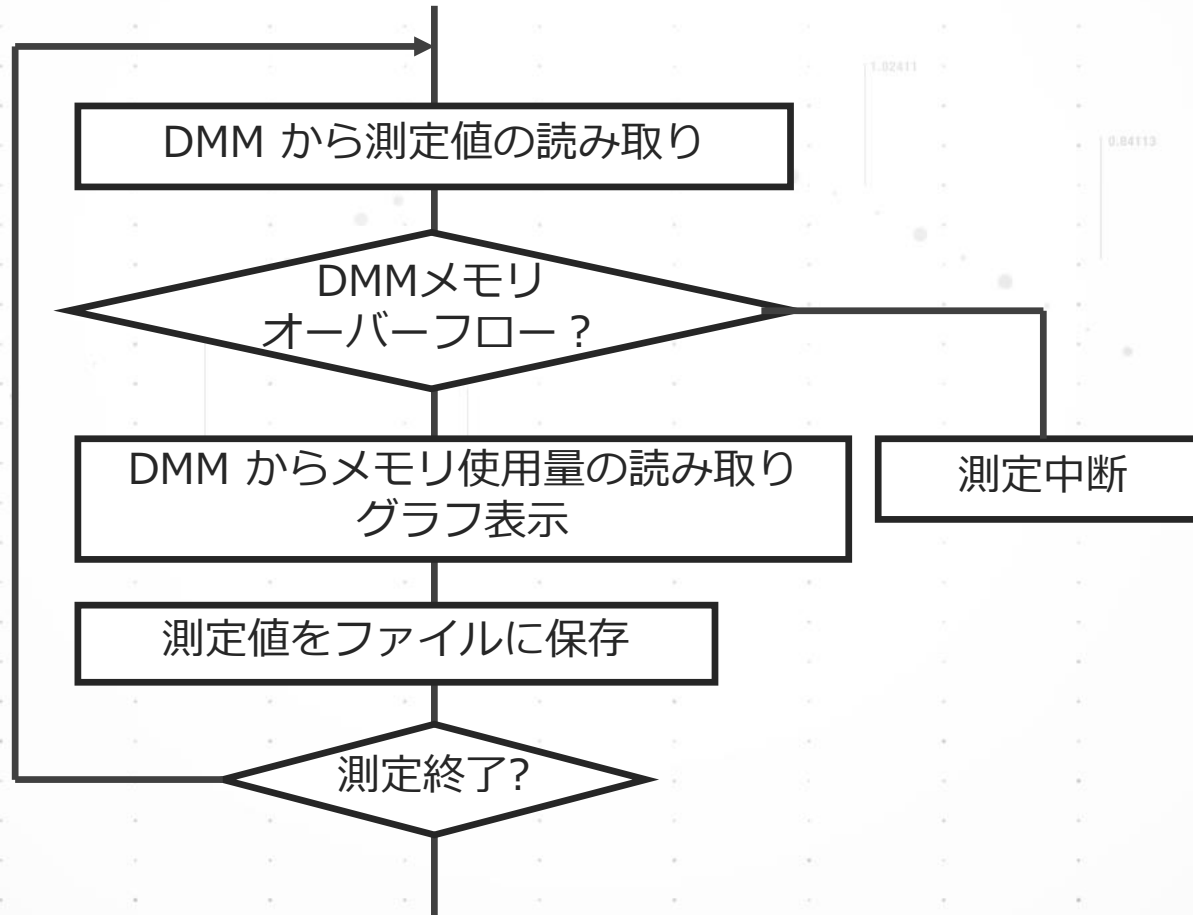
現時点でメモリ使用量の最大値

プログラムの進捗などのメッセージ

デジタルマルチメータ応用例

高速サンプリングかつ長時間測定を実現するアプリケーション

- サンプルプログラム データ転送と保存のフロー



デジタルマルチメータ応用例

高速サンプリングかつ長時間測定を実現するアプリケーション

- サンプルプログラム実行例
- 正常終了

20 μ 秒ごとの測定を
10000 x 9000 回実施

DMM DataLogger Sample program

VISA アドレス: TCPIP0::169.254.4.70::hislip0::INSTR

サンプル数 / 転送: 10000 測定開始

転送回数 / 測定開始ボタン: 9000

サンプリング周期 (秒): 20E-6 測定停止

データログファイル名: DmmDataLog.txt 表示をクリア

DMM Memory

20000
15000
10000
5000
0

0 20 40 60 80 100

転送済みの回数: 9000

DMM メモリの最大使用量: 23904

DMM オープン成功
*IDN?: Keysight Technologies,34470A,MY57701384,A.03.00-02.40-03.00-00.52-04-01
SYST:ERR?: +0, No error
DMM 測定開始 成功
測定完了
SYST:ERR?: +0, No error

LAN HiSLIP で接続

DMMのメモリ使用量は
増える傾向ではない

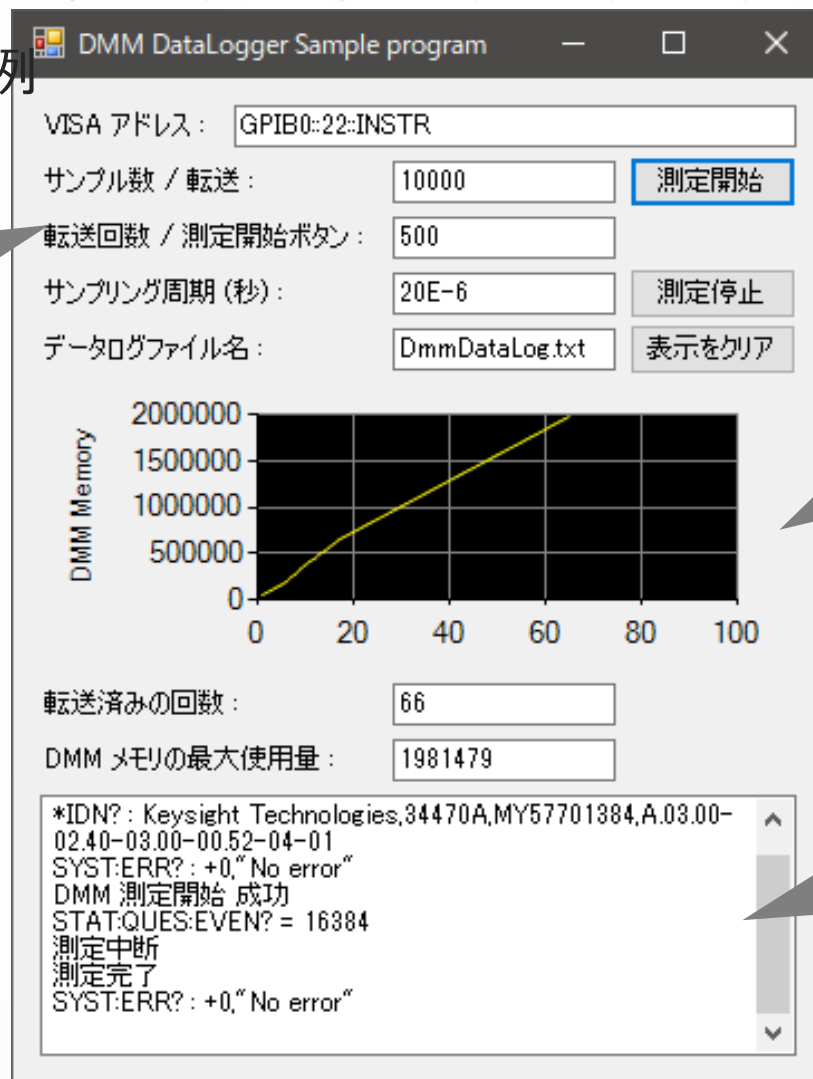
プログラムは正常終了

デジタルマルチメータ応用例

高速サンプリングかつ長時間測定を実現するアプリケーション

- サンプルプログラム実行例
- メモリアーバフロー検出

20μ秒ごとの測定を
10000 × 500 回実施



GPIBで接続
GPIBはLANと比較して
データの転送に時間がかかります

DMMのメモリ使用量は増え続けているので、このままの傾向が続けばメモリがオーバーフローすることが予想できます

DMMのメモリがオーバーフローしたことを検出し、測定を中断
STAT:QUES:EVENT? = 16384

デジタルマルチメータ応用例

高速サンプリングかつ長時間測定を実現するアプリケーション

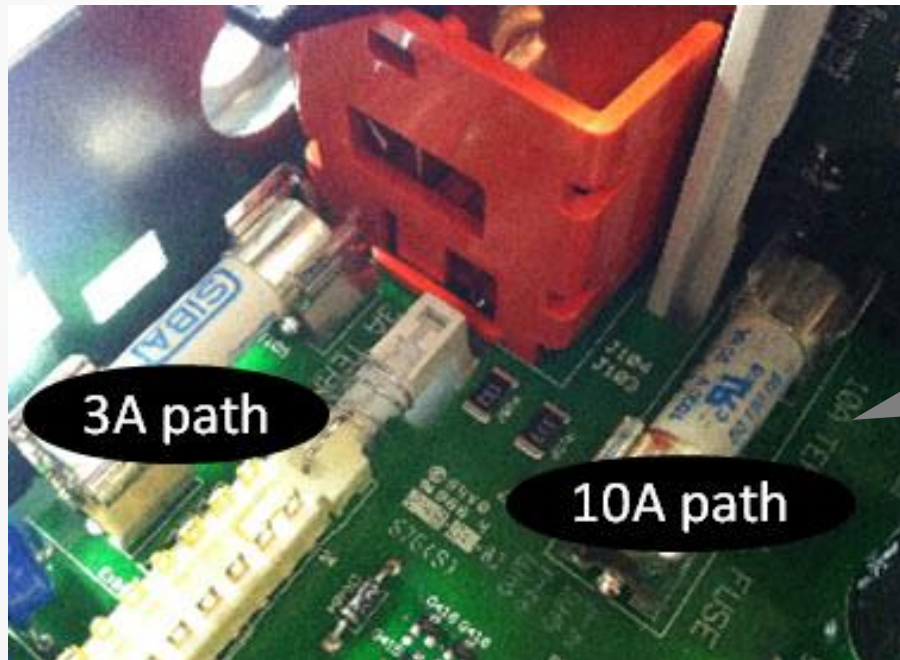
- サンプルプログラム
- 測定部分のソースの一部

```
DmmDataLoggerSample.cs x DmmDataLoggerSample.cs [デザイン]
DmmDataLoggerSample
// データロギング実施のループ
121
122
123 double[] rData; // 測定値を取得する変数
124 int dmmMaxMemory = 0; // ロギング中に使用したDMMの内蔵メモリの最大値を保持する変数
125
126 txtLog.AppendText($"DMM 測定開始 成功\r\n");
127
128 try
129 {
130     for (int i = 0; i < txPerStart; i++)
131     {
132         Application.DoEvents(); // 画面の更新
133         if (StopFlag) // Stopボタンが押されたときの処理
134         {
135             txtLog.AppendText($"測定停止ボタンが検出されました\r\n");
136             dmm.FormattedIO.WriteLine("ABORT"); // 測定を中断
137             break;
138         }
139
140         dmm.FormattedIO.WriteLine($"DATA:REM? {txtSamplePerTx.Text},WAIT"); // データ転送指示
141         rData = dmm.FormattedIO.ReadLineBinaryBlockOfDouble(); // データ受信 (バイナリ)
142         // クエストショナルレジスタでDMMのメモリのオーバーフローが検出できる
143         dmm.FormattedIO.WriteLine("STAT:QUES:EVEN?"); // クエストショナルレジスタの取得
144         short sqe = (short)dmm.FormattedIO.ReadLineInt64();
145         if (sqe != 0) // メモリオーバーフローを含む、DMMの異常を検出
146         {
147             txtLog.AppendText($"STAT:QUES:EVEN? = {sqe}\r\n測定中断\r\n");
148             dmm.FormattedIO.WriteLine("ABORT"); // 測定を中断
149             break;
150         }
151         //
152         // DMMの測定速度より、データ転送速度が遅い場合、DMMのメモリ内のデータ数が増加する
153         // DMMのメモリ内のデータ数により、今後、オーバーフローするかどうかが予想できる
154         //
155         dmm.FormattedIO.WriteLine("DATA:POIN?"); // DMMのメモリ内のデータの個数を問い合わせ
156         int dp = (int)dmm.FormattedIO.ReadLineInt64();
157         dmmMaxMemory = dp > dmmMaxMemory ? dp : dmmMaxMemory; // データ個数の最大値を保持
158         txtMaxDmmMem.Text = dmmMaxMemory.ToString();
159         chtDmmMem.Series[0].Points.Add(dp); // データの個数をグラフ表示
160         if ((i % 100) == 0) chtDmmMem.Series[0].Points.Clear(); // 100個おきにグラフを更新
161
162         // 測定値をファイルに追記で保存する
163         using (var writer = new System.IO.StreamWriter(txtFileName.Text, append: true))
164         {
165             foreach (var r in rData)
166             {
167                 var s = r.ToString("0.0000E+00");
168                 writer.WriteLine(s);
169             }
170         }
171         txtNumberOfTxd.Text = (i + 1).ToString(); // データ転送回数の表示
172     }
173 }
174 catch (Exception ex)
175 {
176     txtLog.AppendText($"DMM Data Logging Loop failed : {ex.Message}\r\n");
177     return;
178 }
---
```

デジタルマルチメータ Q&A

よくあるご質問

- Q: 電流測定を行っているが、異常な測定値が表示される
- A: DMM の電流測定経路には、過電流保護のためのヒューズが使用されています。このヒューズが切れた場合、DMMはヒューズ切れを検出できませんので、実際には測定できていない電流値を表示します。

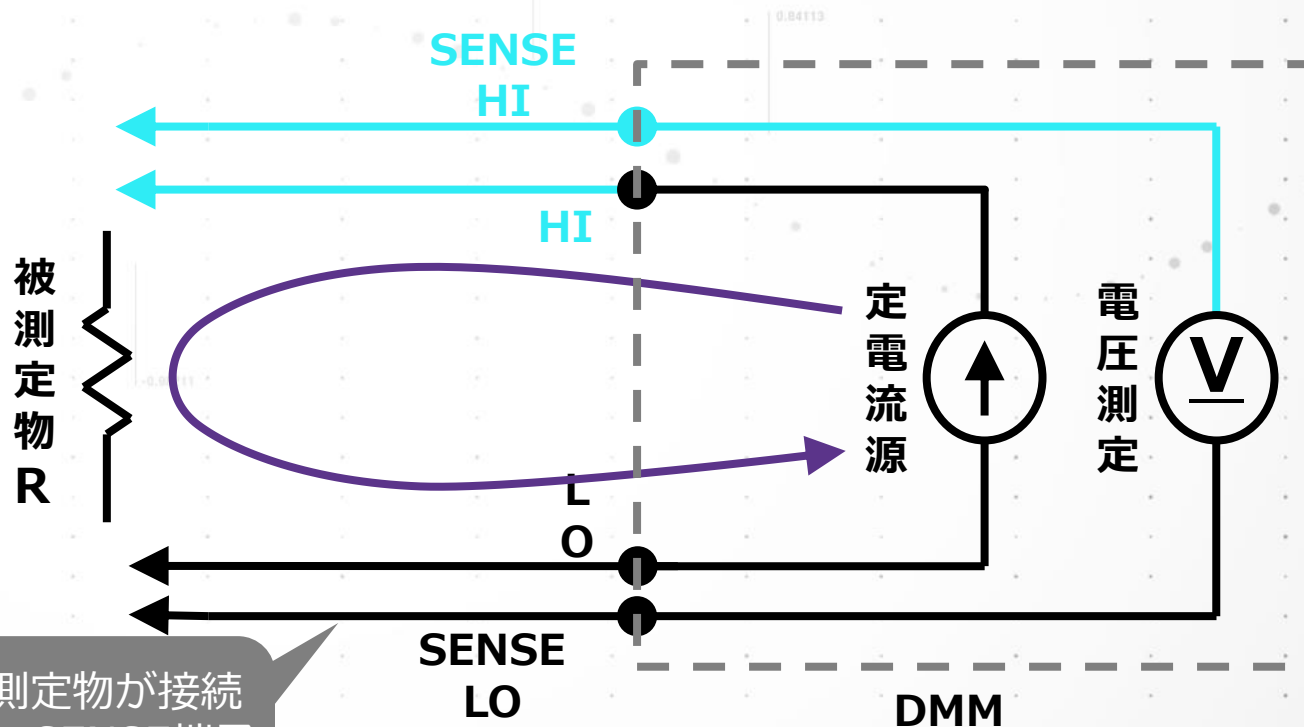


機種によっては過電流ほぼフューズは内部の基板に実装されています。

デジタルマルチメータ Q&A

よくあるご質問

- Q: 抵抗測定で何も接続していない状態でも抵抗値が表示される
- A: 2線式抵抗測定の場合、何も接続していない状態ではオーバーロードが表示されますが、4線式抵抗測定の場合は、センス端子に発生する電圧に相当する抵抗値を表示しますのでオーバーロードにならない場合があります。



4線式抵抗測定の場合被測定物が接続されていないと電圧測定のSENSE端子に開放端電圧が発生します。

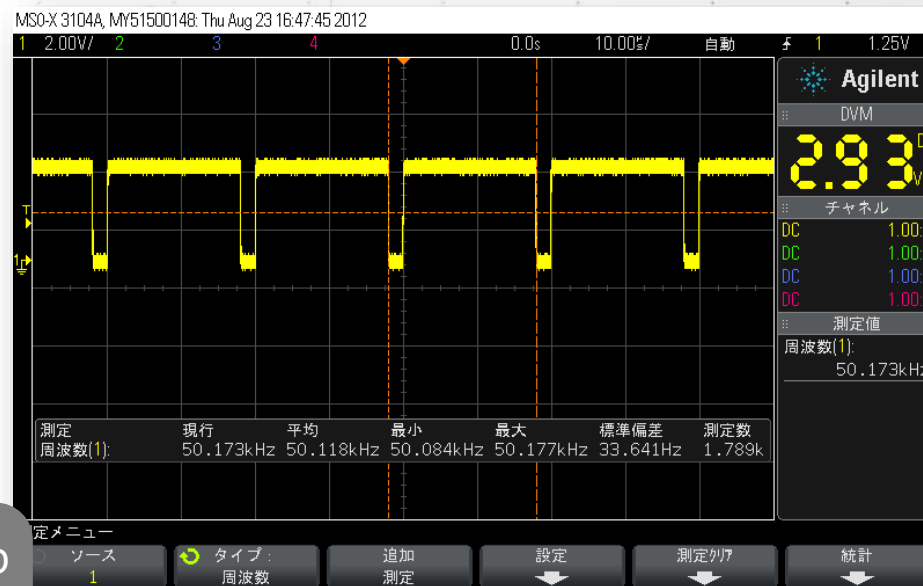
デジタルマルチメータ Q&A

よくあるご質問

- Q: 34465A を使用して自動計測を行っている。測定時間がたまにばらつく
- A: 34465A はビットマップディスプレイを採用しており、トレンドチャートなどの表示も可能です。一方、測定値を更新するためのオーバーヘッドも大きいいため、測定が遅延する場合があります。この遅延を避けるには、DISP OFF コマンドでディスプレイ表示を OFF にします。



キーサイトのDMMには測定終了を知らせるVM Compという信号が出力されています。その信号を測定することで測定時間を知ることができます。



デジタルマルチメータ Q&A

よくあるご質問

<https://www.keysight.com/main/support.jspx?cc=US&lc=eng&jmpid=zzkeysupport>

KEYSIGHT TECHNOLOGIES

ハードウェア ソフトウェア サービス&サポート 業種&テクノロジー キーサイトについて myKeysight

日本 ホーム > Services & Support お問い合わせ窓口

サポート

製品番号で検索: 検索

よくある質問 (FAQ)

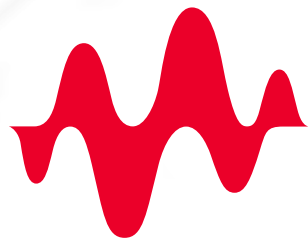
KeysightCare 修理サービス

さまざまな製品購入方法 テクノロジーリフレッシュサービス

Test as a Service (TaaS) コンサルティングサービス

技術資料 ライブラリ ドライバ/ファームウェア/ソフトウェア

トレーニング、イベント ソフトウェア・ライセンス ディスカッション・フォーラム



KEYSIGHT
TECHNOLOGIES

4.50221