

```

1  using System;
2  using System.IO;
3  using System.Windows.Forms;
4  using Ivi.Visa.Interop;          //VISAライブラリ 名前空間指定。
5
6  // KI-VISAを使った計測器 (電源・負荷装置等) 制御サンプルコード。
7  // Copyright by Kikusui Electronics Corp. 2015-2022
8  // コード作成:SE2課 矢島。
9
10 // 2015.12.22 Ver1.00 初版。
11 // 2016.01.27 Ver1.01 コメント文の追加。
12 // TerminationCharacterEnabledの設定をデフォルトのfalseからtrueに変更。
13 // シリアルポーリングできないインタフェースではSerialPollボタンを無効にした。
14 // 2017.01.11 Ver1.02 証明書ストアの問題でビルドできないことがあったので、ClickOnce署名のチェックを無効にした。
15 // 2018.05.16 Ver1.03 NI-VISA 17.0でリソースマネージャOpenに失敗するため、引数の内容を変更した。
16 // 2020.06.01 Ver1.04 FindResourceとOpenメソッド、クエリメッセージ受信処理にTry～Catchを追加。
17 // 2020.08.27 Ver1.05 コメント、Aboutボックスの表記内容修正。
18 // 2020.10.15 Ver1.10 Web公開リリース版。
19 // 2022.05.09 Ver1.11 Timeout指定の方法を修正した。
20
21 namespace SampleCode_KIVISA
22 {
23     public partial class Form1 : Form
24     {
25         //リソースマネージャオブジェクト作成。
26         IResourceManager3 rm;
27         IMessage msg;
28
29         //ダイアログボックスのタイトル。
30         readonly string sAppTitle = "KI-VISA Sample for CSharp";
31
32         //メインフォーム。
33         public Form1 ()
34         {
35             InitializeComponent();
36             //フォーム表示位置を画面センターに指定。
37             this.StartPosition = FormStartPosition.CenterScreen;
38         }
39
40         //フォーム表示時の処理。
41         private void Form1_Load(object sender, EventArgs e)
42         {
43
44             //FindResourceで見つかったVISAリソースリスト格納変数。
45             string[] foundResources;
46
47             //リソースマネージャを初期化。
48             rm = new ResourceManager();
49
50             //FindResourceで使用可能なINSTRを含むVISAリソース名を取得する。
51             //詳細はKI-VISA Library VISA COM ガイドブック 8- VISAアドレスの動的な設定 (48ページ以降)
52             //をご覧ください。
53             try
54             {
55                 foundResources = rm.FindRsrc("?*INSTR");
56             }
57             catch (Exception)
58             {
59                 MessageBox.Show("使用可能なVISAリソースがコンピュータに1つも見つかりません。" + "\r"
60                     + "デバイスが接続されているか確認してください",
61                     sAppTitle, MessageBoxButtons.OK, MessageBoxIcon.Stop);
62                 return;
63             }
64
65             //VISA Resource名のコンボボックスに取得したVISAリソース名を追加する。
66             cmb_Resources.DataSource = foundResources;
67
68             //設定コマンドをコンボボックスに追加する。
69             //リストにないコマンドは、コンボボックスですので実行時に手入力可能です。
70             //ここのコマンドを編集すれば任意のコマンドをプルダウンメニューから選択できるようになります。
71             //代表例として、PMX18-5A (直流電源 18V/5Aモデル) で使用可能なコマンドを記載しています。
72

```

```

73 cmb_CommandString.Items.Add("SYST:REMOTE");
74 cmb_CommandString.Items.Add("SYST:LOCAL");
75 cmb_CommandString.Items.Add("*CLS");
76 cmb_CommandString.Items.Add("OUTP ON");
77 cmb_CommandString.Items.Add("OUTP OFF");
78 cmb_CommandString.Items.Add("VOLT 10");
79 cmb_CommandString.Items.Add("VOLT 1");
80 cmb_CommandString.Items.Add("CURR 1");
81 cmb_CommandString.Items.Add("CURR 0.1");
82 cmb_CommandString.SelectedIndex = 0; // 最初のコマンドを表示させる。
83
84 // クエリコマンドをコンボボックスに追加する。
85 cmb_QueryString.Items.Add("*IDN?");
86 cmb_QueryString.Items.Add("*STB?");
87 cmb_QueryString.Items.Add("SYST:ERR?");
88 cmb_QueryString.Items.Add("OUTP?");
89 cmb_QueryString.Items.Add("VOLT?");
90 cmb_QueryString.Items.Add("CURR?");
91 cmb_QueryString.Items.Add("READ:VOLT?");
92 cmb_QueryString.Items.Add("READ:CURR?");
93 cmb_QueryString.SelectedIndex = 0; // 最初のクエリコマンドを表示させる。
94
95 // VISAセッションをオープンしない段階ではボタンをDisableにする。
96 btn_SendCommand.Enabled = false;
97 btn_SendQuery.Enabled = false;
98 btn_SerialPolling.Enabled = false;
99
100 // -----
101 // [Info] LANデバイスのサーチについて。
102 // LANデバイスについてはダイナミックサーチが使えないことがあります。
103 // その際はKI-VISA Instruments ExplorerでFindすると使えるようになります。
104 // -----
105 }
106
107 // VISAセッションを開く。
108 private void btn_Open_Click(object sender, EventArgs e)
109 {
110     // 詳細はKI-VISA Library VISA COM ガイドブック 6- VISAセッションのオープン(35
111     // ページ以降)をご覧ください。
112     IGpib gpib;
113     ISerial seri;
114     ITcpipInstr tcp;
115     IUsb usb;
116
117     // コンボボックスで指定したリソース名のVISAセッションを開く。
118     try
119     {
120         msg = (IMessage)rm.Open(cmb_Resources.Text,
121             AccessMode.EXCLUSIVE_LOCK, 2000); // 2022.5.9
122         // Timeout指定方法を修正(NO__LOCK→EXCLUSIVE_LOCK 2000=2秒) yajima
123         MessageBox.Show("VISAセッションオープンに成功しました" + "\r" +
124             cmb_Resources.Text, sAppTitle, MessageBoxButtons.OK,
125             MessageBoxIcon.Asterisk);
126     }
127     catch (Exception)
128     {
129         MessageBox.Show("VISAセッションオープンに失敗しました" + "\r" +
130             cmb_Resources.Text, sAppTitle, MessageBoxButtons.OK,
131             MessageBoxIcon.Stop);
132         return;
133     }
134
135     // インタフェースタイプ別設定。
136     // 詳細はKI-VISA Library VISA COM ガイドブック 7- 基本的なIO(39ページ以降)をご覧ください。
137     msg.TerminationCharacter = 10;
138     msg.TerminationCharacterEnabled = true; // IEEE488.2
139     // 計測器でReadString()で正しくターミネートするためにはtrue推奨。
140     msg.SendEndEnabled = true;
141
142     // HardwareInterfaceTypeによってインタフェースタイプ別設定を行う。
143     switch (msg.HardwareInterfaceType)
144     {
145         case 1:

```

```

138         //GPIB
139         gpib = (IGpib)msg;
140         gpib.RepeatAddressingEnabled = true;
141         gpib.UnaddressingEnabled = false;
142
143         //シリアルポーリングボタンを有効にする (GPIB, USBのみ使用可能なため).
144         btn_SerialPolling.Enabled = true;
145
146         break;
147
148     case 4:
149         //ASRL (シリアルポート).
150         //ボーレートの設定はこのパラメータを変えて行います.
151         seri = (ISerial)msg;
152         seri.BaudRate = 19200;
153         //ボーレート 19200bps.
154         seri.DataBits = 8;
155         //データビット 8ビット.
156         seri.StopBits = SerialStopBits.ASRL_STOP_ONE;
157         //ストップビット 1ビット.
158         seri.Parity = SerialParity.ASRL_PAR_NONE; //パリティなし.
159         seri.FlowControl = SerialFlowControl.ASRL_FLOW_NONE;
160         //フロー制御なし.
161
162         //シリアルポーリングボタンを無効にする.
163         btn_SerialPolling.Enabled = false;
164
165         break;
166
167     case 6:
168         //LAN
169         //注意: LANデバイスについてはダイナミックサーチが使えないことがあります.
170         //その際はKI-VISA Instruments ExplorerでFindすると使えるようになります.
171         tcp = (ITcpipInstr)msg;
172
173         //シリアルポーリングボタンを無効にする.
174         btn_SerialPolling.Enabled = false;
175
176         break;
177
178     case 7:
179         //USB
180         usb = (IUsb)msg;
181
182         //シリアルポーリングボタンを有効にする (GPIB, USBのみ使用可能なため).
183         btn_SerialPolling.Enabled = true;
184
185         break;
186     }
187
188     //ボタンをEnableにする.
189     btn_SendCommand.Enabled = true;
190     btn_SendQuery.Enabled = true;
191
192     }
193
194     //コマンド送信.
195     private void btn_writeString_Click(object sender, EventArgs e)
196     {
197         int r;
198
199         //cmb_CommandStringで選んだコマンドを送信.
200         r = msg.WriteString(cmb_CommandString.Text+"\n"); //コマンドの後デリミタが必要.
201
202         //直接コマンドを書きたい場合はこのようにも書けます.
203         //r = msg.WriteString("VOLT 10\n");
204         //r = msg.WriteString("OUTP ON\n");
205     }
206
207     //クエリコマンド送信・リザルト受信の例.
208     private void btn_SendQuery_Click(object sender, EventArgs e)
209     {
210         int r;

```

```

207         string strret = "";
208
209         //クエリコマンド送信.
210         r = msg.WriteString(cmb_QueryString.Text + "\n"); //コマンドの後デリミタが必要.
211
212         //クエリのリザルトを受信.
213         try
214         {
215             strret = msg.ReadString(1024); // 1024バイトバッファ.
216         }
217         catch (Exception)
218         {
219             MessageBox.Show("クエリメッセージ受信に失敗しました"
220                 + "\r" + cmb_Resources.Text,
221                 sAppTitle, MessageBoxButtons.OK, MessageBoxIcon.Stop);
222             return;
223         }
224
225
226         //ReadStringした値を返す.
227         tb_Result.Text = strret;
228     }
229
230     //ステータスバイトレジスタをシリアルポーリングする例.
231     private void btn_SerialPolling_Click(object sender, EventArgs e)
232     {
233         short stb;
234
235         //ステータスバイトレジスタをシリアルポーリング.
236         //ReadSTBはGPIOB, USBインタフェースのみ使用可能です.
237         //他のインタフェースでは実行すると例外発生となりますのでボタンを無効にしています.
238         stb = msg.ReadSTB();
239
240         //リザルトをテキストボックスに書き込み.
241         tb_Result.Text = stb.ToString();
242     }
243
244     //フォームを閉じたときの処理.
245     private void quitToolStripMenuItem_Click(object sender, EventArgs e)
246     {
247         //VISAセッションを閉じる.
248         if (msg != null)
249         {
250             msg.Close();
251         }
252         //フォームを閉じる.
253         this.Close();
254     }
255
256     //Resultテキストボックスをクリアする.
257     private void btn_ClearResult_Click(object sender, EventArgs e)
258     {
259         //Resultテキストボックスをクリアする.
260         tb_Result.Text = "";
261     }
262
263     //AboutBox バージョン表示.
264     private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
265     {
266         MessageBox.Show(
267             "Samplecode for KI-VISA CSharp Ver1.11" + "\r"
268             + "Copyright (C) 2020 Kikusui Electronics Corp." + "\r",
269             sAppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
270     }
271 }
272
273
274
275 }
276

```