

```
1 Imports System.Threading
2 Imports System.Windows.Forms
3 ''' <summary>
4 ''' TOS6200Aシリーズ用 試験データ取得サンプルコード メインフォーム
5 ''' 2022.02.01 Ver1.00 SE 矢島
6 '''
7 ''' </summary>
8 Public Class frmmain
9
10     Dim Tos6200Func As New ClsTos6200           'TOS6200シリーズ用関数モジュール
11     Dim Tos6200 As New ClsVisaCommunication      'TOS6200 VISA通信共通モジュール
12     Dim file As New ClsFileAccess              'ファイルアクセス共通モジュール
13     Dim cancelTasks As New CancellationTokensource 'タスクキャンセル用トークン
14
15     '-----
16     'フォームイベントハンドラ
17     '-----
18     ''' <summary>
19     ''' メインフォーム読み込み時の処理
20     ''' </summary>
21     ''' <param name="sender"></param>
22     ''' <param name="e"></param>
23     Private Sub frmMain_Load(sender As Object, e As EventArgs) Handles MyBase.Load
24
25         'フォーム名を設定
26         Me.Text = My.Application.Info.AssemblyName + " Version " + Convert.ToString(My.Application.Info.Version)
27
28         'フォームオープン時の処理
29         AppStart() 'フォームオープン時の処理
30
31     End Sub
32
33     ''' <summary>
34     '''
35     ''' メインフォームクローズ時の処理
```

```
36     ''' </summary>
37     ''' <param name="sender"></param>
38     ''' <param name="e"></param>
39     Private Sub frmMain_Closed(sender As Object, e As EventArgs) Handles Me.Closed
40
41         AppClose()
42
43     End Sub
44
45     ''' <summary>
46     ''' 開始ボタンの処理(非同期処理関数呼び出し)
47     ''' </summary>
48     ''' <param name="sender"></param>
49     ''' <param name="e"></param>
50     Private Sub btn_TestStart_Click(sender As Object, e As EventArgs) Handles btn_TestStart.Click
51
52         '非同期実行でTOS6200の状態をモニターし、試験結果をファイルに保存する
53         Async_TestExec()
54
55     End Sub
56
57     ''' <summary>
58     ''' 停止ボタンの処理
59     ''' </summary>
60     ''' <param name="sender"></param>
61     ''' <param name="e"></param>
62     Private Sub btn_MonitorStop_Click(sender As Object, e As EventArgs) Handles btn_TestStop.Click
63
64         '非同期実行のタスクキャンセルトークンを発行する
65         cancelTasks.Cancel()
66
67     End Sub
68
69     ''' <summary>
70     ''' 接続ボタンの処理
71     ''' </summary>
72     ''' <param name="sender"></param>
```

```
71     ''' <param name="e"></param>
72     Private Sub btn_Connect_Click(sender As Object, e As EventArgs) Handles btn_Connect.Click
73
74         ConnectToTOS()
75
76     End Sub
77
78     ''' <summary>
79     ''' リソース検索ボタンの処理
80     ''' </summary>
81     ''' <param name="sender"></param>
82     ''' <param name="e"></param>
83     Private Sub btn_FindResource_Click(sender As Object, e As EventArgs) Handles btn_FindResource.Click
84
85         cmb_VisaResourceName.DataSource = Tos6200.VisaFindResouerce 'VISAリソースを検索する
86
87     End Sub
88     ''' <summary>
89     ''' 試験データファイル保存パス名取得ボタンの処理
90     ''' </summary>
91     ''' <param name="sender"></param>
92     ''' <param name="e"></param>
93     Private Sub btn_GetSaveFilePath_Click(sender As Object, e As EventArgs) Handles btn_GetSaveFilePath.Click
94
95         GetSaveFilePath()
96
97     End Sub
98
99     ''' <summary>
100    ''' メモリ読み出しボタンの処理
101    ''' </summary>
102    ''' <param name="sender"></param>
103    ''' <param name="e"></param>
104    Private Sub btn_RecallMemory_Click(sender As Object, e As EventArgs) Handles btn_RecallMemory.Click
105
```

```
106     RecallMemory()
107
108 End Sub
109
110     ''' <summary>
111     ''' メニューバー 切断コマンドの処理
112     ''' </summary>
113     ''' <param name="sender"></param>
114     ''' <param name="e"></param>
115 Private Sub ToolStripMenuItem_Disconnect_Click(sender As Object, e As EventArgs) Handles ToolStripMenuItem_Disconnect.Click
116
117     DisconnectToTOS()
118
119 End Sub
120
121     ''' <summary>
122     ''' メニューバー 終了コマンドの処理
123     ''' </summary>
124     ''' <param name="sender"></param>
125     ''' <param name="e"></param>
126 Private Sub ToolStripMenuItem_Exit_Click(sender As Object, e As EventArgs) Handles ToolStripMenuItem_Exit.Click
127
128     AppClose()
129     Application.Exit()
130
131 End Sub
132
133     ''' <summary>
134     ''' メニューバー ヘルプコマンドの処理
135     ''' </summary>
136     ''' <param name="sender"></param>
137     ''' <param name="e"></param>
138 Private Sub ToolStripMenuItem_Version_Click(sender As Object, e As EventArgs) Handles ToolStripMenuItem_Version.Click
139
```

```
140 '-----
141 'バージョン表示ダイアログを開く
142 '-----
143     MessageBox.Show(Convert.ToString(My.Application.Info.AssemblyName) + " Version " +
144         Convert.ToString(My.Application.Info.Version) + vbCrLf +
145         My.Application.Info.Copyright, "バージョン情報")
146 End Sub
147
148
149 '-----
150 'プライベート関数(フォーム内の処理で使います)
151 '-----
152 ''' <summary>
153 '''   プログラム開始時の処理
154 ''' </summary>
155 Private Sub AppStart()
156
157     'メインのタブを選択する
158     TabControl1.SelectTab(0)
159
160     'データ保存回数表示を0にリセットする
161     tb_RecCount.Text = 0
162
163     '画面更新タイマーを設定
164     Timer1.Enabled = False      '更新停止
165     Timer1.Interval = 100      '更新周期(ms)
166
167     '接続状態を表示する
168     tb_Result.Text = "未接続"
169
170     'モニター開始、停止ボタンをDisableにする
171     btn_TestStart.Enabled = False
172     btn_TestStop.Enabled = False
173
174     'メモリ読み出しボタン 無効にする
```

```
175     btn_RecallMemory.Enabled = False
176
177     'FindResourceでVISAリソースをコンボボックスに読み込む
178     cmb_VisaResourceName.DataSource = Tos6200.VisaFindResource
179
180     '設定に保存されているVISAアドレスが空でなければコンボボックスに前回のVISAアドレスを入れる
181     If My.Settings.VISA_Resouce <> "" Then
182         cmb_VisaResourceName.Text = My.Settings.VISA_Resouce
183     End If
184
185     '設定に保存されているフォルダパスが空でなければフォルダパスのテキストボックスにパス名を入れる
186     If My.Settings.DataFolder <> "" Then
187         tb_DataSaveFolderName.Text = My.Settings.DataFolder
188     End If
189
190     'フォルダパスのテキストボックスが空だった場合、初期値としてMyDocumentsフォルダを指定する
191     If tb_DataSaveFolderName.Text = "" Then
192         tb_DataSaveFolderName.Text = System.Environment.GetFolderPath(Environment.SpecialFolder.Personal)
193     End If
194
195     '設定に保存されているフォルダパスが空でなければフォルダパスのテキストボックスにパス名を入れる
196     If My.Settings.DataFileName <> "" Then
197         tb_DataFileName.Text = My.Settings.DataFileName
198     End If
199
200     '設定に保存されているメモリ番号をテキストボックスに入れる
201     tb_MemNo.Text = My.Settings.MemoryNo
202
203 End Sub
204
205 ''' <summary>
206 ''' プログラム終了時の処理
207 ''' </summary>
208 Private Sub AppClose()
209
```

```
210     MessageBox.Show("プログラムを終了します", My.Application.Info.AssemblyName, MessageBoxButtons.OK,  
        MessageBoxIcon.Information)  
211  
212     '設定を保存する  
213     My.Settings.VISA_Resouce = cmb_VisaResourceName.Text  
214     My.Settings.DataFolder = tb_DataSaveFolderName.Text  
215     My.Settings.DataFileName = tb_DataFileName.Text  
216  
217     'アプリケーションの設定を保存する  
218     ' 設定の保存先 "C:\Users\(ユーザー名)\AppData\Local\SampleCode_TOS6200A_Serie" 配下の  
219     ' user.config に記録されています。  
220     My.Settings.Save()  
221  
222     '画面更新のタイマーを停止する  
223     Timer1.Enabled = False  
224  
225     'VISAセッションをクローズする  
226     Tos6200.VisaClose()  
227  
228 End Sub  
229 ''' <summary>  
230 ''' 測定開始/停止ボタンの状態を設定する  
231 ''' </summary>  
232 ''' <param name="Status"></param>  
233 Private Sub ControlButtonStatus(Status As Boolean)  
234  
235     If Status = True Then  
236         'ボタンの表示  
237         btn_TestStart.Text = "試験中"  
238         btn_TestStop.Text = "試験停止"  
239  
240         'ステータスラベルの表示  
241         ts_StatusLabel1.Text = "試験中"  
242  
243         'ボタン/テキストボックスの有効/無効
```

```
244         '有効
245         btn_TestStop.Enabled = True
246
247         '無効
248         btn_TestStart.Enabled = False
249         btn_RecallMemory.Enabled = False
250         tb_MemNo.Enabled = False
251
252     Else
253         'ボタンの表示
254         btn_TestStart.Text = "試験開始"
255         btn_TestStop.Text = "準備完了"
256
257         'ステータスラベルの表示
258         ts_StatusLabel.Text = "準備完了"
259
260         'ボタン/テキストボックスの有効/無効
261         '有効
262         btn_TestStart.Enabled = True
263         btn_RecallMemory.Enabled = True
264         tb_MemNo.Enabled = True
265
266         '無効
267         btn_TestStop.Enabled = False
268
269     End If
270
271 End Sub
272 ''' <summary>
273 ''' GUI更新タイマー
274 ''' </summary>
275 ''' <param name="sender"></param>
276 ''' <param name="e"></param>
277 Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
278
```



```
279         tb_Result.Text = Tos6200Func.TestStatus 'ステータスプロパティを表示
280
281     End Sub
282
283     ''' <summary>
284     ''' TOS6200 試験状態のモニターと試験結果取得(フォーム依存、非同期動作)
285     ''' </summary>
286     Private Async Sub Async_TestExec()
287
288         cancelTasks = New CancellationTokencSource() 'タスクキャンセルトークンを初期化
289
290         Dim sResult As String
291         Dim sDataSavepath As String '保存先パス名
292         Dim sHeader As String 'CSVファイルの項目名
293         Dim sDateTime As String '試験開始日時
294         Dim rd As DialogResult 'Messageboxの応答
295
296         '保存先フォルダが空白だったときは関数を終了
297         If tb_DataSaveFolderName.Text = "" Then
298             rd = MessageBox.Show("保存先フォルダが指定されていません。" + vbCrLf +
299                 "[参照]ボタンをクリックして保存先フォルダを選択してください",
300                 "エラー",
301                 MessageBoxButtons.OK,
302                 MessageBoxIcon.Error)
303             Exit Sub
304         End If
305
306         'メモリ読み出し状態の確認ダイアログ
307         If tb_MemName.Text = "" Then
308
309             'メモリ名の値が空白だったとき
310             rd = MessageBox.Show("メモリの読み出しがされていません。" + vbCrLf +
311                 "このまま試験を開始しますか？",
312                 "確認",
313                 MessageBoxButtons.YesNo,
```

```
314         MessageBoxIcon.Question)
315
316     'NOが押されていたら関数を終了
317     If rd = vbNo Then
318         MessageBox.Show("メモリ番号を指定して[読出]ボタンをクリック後、試験を開始してください",
319             "メモリを読み出してください",
320             MessageBoxButtons.OK,
321             MessageBoxIcon.Information)
322     Exit Sub
323
324     End If
325 End If
326
327 'GUI更新を開始する
328 Timer1.Enabled = True
329
330 'ボタンステータスを「試験中」にする
331 ControlButtonStatus(True)
332
333 Try
334     '測定開始日時を取得する
335     sDateTime = DateTime.Now.ToString("yyyy,MM,dd,HH,mm,ss")
336
337     '試験を実行する
338     Tos6200.VisaSendCommand("SIL 1")    'エコーバック停止
339     Tos6200.VisaSendCommand("START")    '試験開始
340
341     '状態を監視し、試験が終了したら結果を取得する
342     sResult = Await Task.Run(Function() Tos6200Func.GetResult_ASync(Tos6200, cancelTasks.Token))
343
344     'ボタンを「試験開始待ち状態」にする
345     ControlButtonStatus(False)
346
347     '応答がCancelやErrorの場合は関数を抜ける
348     If sResult = "Cancel" Or sResult = "Error" Then
```

```
349         ControlButtonStatus(False)           'ボタンを「試験開始待ち」にする
350         Exit Sub
351
352     End If
353
354     '取得した測定データをフォーマットし、日時やメモリ名、シーケンス番号を追加する
355     sResult = Tos6200Func.EditResults(Tos6200, sResult, sDateTime, Integer.Parse(tb_MemNo.Text))
356
357     '結果ファイルの保存先
358     sDataSavepath = tb_DataSaveFolderName.Text + "\" + tb_DataFileName.Text
359
360     'ファイルが新規作成の場合、先頭行に凡例を入れる
361     If file.FileExist(sDataSavepath) = False Then
362         sHeader = "試験番号,-,-,試験開始(年),(月),(日),試験開始(時),(分),(秒),判定時電圧値 (V) ,電流値 (A) ,抵抗値 (Ω),試
        験時間,判定結果,限度値(Ω),周波数(Hz),メモリ番号 ,メモリ名"
363         file.WriteAllTextToFile(sDataSavepath, sHeader, False)
364     End If
365
366     '試験結果を保存する
367     file.WriteAllTextToFile(sDataSavepath, sResult, True)
368
369     'アプリ起動時からの保存数表示を+1する
370     tb_RecCount.Text = Integer.Parse(tb_RecCount.Text) + 1
371
372     'FailやProtectionの場合の処理
373     If Tos6200Func.DsrValue = 32 Or Tos6200Func.DsrValue = 128 Then           'FailかProtectionの場合
374         MessageBox.Show("判定は" + Tos6200Func.TestStatus + "です。OKをクリックするとブザーを停止します", "TOS6200
        sample")
375         Tos6200.VisaSendCommand("STOP")           'ストップ
376     End If
377
378     '試験シリアル番号がint32の上限になったら
379     If My.Settings.TestNumber >= Int32.MaxValue Then
380         My.Settings.TestNumber = 1           '1にクリアする
381     Else
```

```
382         '試験シリアル番号をカウントアップ
383         My.Settings.TestNumber = My.Settings.TestNumber + 1
384     End If
385
386     '試験が終了したのでTOSをLOCAL状態に戻す
387     Tos6200.VisaSendCommand("LOC")
388
389     Catch ex As Exception
390         MessageBox.Show("Async_TestExec関数実行時にエラーが発生しました " + vbCrLf +
391             "エラー詳細" + vbCrLf +
392             ex.ToString,
393             My.Application.Info.AssemblyName,
394             MessageBoxButtons.OK,
395             MessageBoxIcon.Error)
396
397         Tos6200.VisaSendCommand("STOP")    'ストップ
398         Tos6200.VisaSendCommand("LOC")    'ローカル
399         ConnectButtonEnable(True)         '接続ボタンを有効に戻す
400         btn_RecallMemory.Enabled = False 'メモリ読み出しボタン 無効
401     Exit Sub
402
403 End Try
404
405 End Sub
406
407 ''' <summary>
408 ''' 接続処理(フォーム依存)
409 ''' </summary>
410 Private Sub ConnectToTOS()
411
412     Dim sVisaAddress As String
413     sVisaAddress = cmb_VisaResourceName.Text    '接続するVISAアドレスをコンボボックスから取得する
414
415     '-----
416     '接続処理
```

```
417 '-----
418 Try
419     Tos6200.VisaOpen(sVisaAddress)          '接続する
420
421 Catch ex As Exception
422     MessageBox.Show("TOS6200シリーズとの接続に失敗しました" & vbCrLf &
423         "VISAアドレスの間違いがないか、" & vbCrLf &
424         "機器の電源がONになっているか等をご確認ください",
425         My.Application.Info.AssemblyName, MessageBoxButtons.OK, MessageBoxIcon.Error)
426
427 Exit Sub
428
429 End Try
430
431 '-----
432 '接続成功後の処理
433 '-----
434 My.Settings.VISA_Resouce = sVisaAddress      '接続に成功したVISAアドレスを設定に保存する
435 ConnectButtonEnable(False)                  '接続関係のボタンを無効にする
436
437 tb_Result.Text = "接続完了"
438 ts_StatusLabel.Text = "接続完了"
439 btn_RecallMemory.Enabled = True              'メモリ読み出しボタン 有効
440
441 'ボタンステータスを「モニター停止中」にする
442 ControlButtonStatus(False)
443
444 End Sub
445
446 ''' <summary>
447 ''' 切断処理(フォーム依存)
448 ''' </summary>
449 Private Sub DisconnectToTOS()
450
451     '-----
452     '画面更新タイマーが動作していたら停止する
```

```
452 '-----
453 If Timer1.Enabled = True Then
454     Timer1.Enabled = False
455     Thread.Sleep(1) '更新タイマーの停止を1秒待つ
456 End If
457
458 '-----
459 '切断処理
460 '-----
461 Try
462     Tos6200.VisaClose()
463     MessageBox.Show("切断しました",
464         My.Application.Info.AssemblyName, MessageBoxButtons.OK, MessageBoxIcon.Information)
465
466 Catch ex As Exception
467     MessageBox.Show("切断に失敗しました。アプリケーションソフトを再起動してください",
468         My.Application.Info.AssemblyName, MessageBoxButtons.OK, MessageBoxIcon.Information)
469     Exit Sub
470
471 End Try
472
473 '-----
474 '切断成功後の処理
475 '-----
476
477 '接続ボタンを有効にする
478 ConnectButtonEnable(True)
479
480 'テキストボックスの表示
481 tb_Result.Text = "未接続"
482 ts_StatusLabel.Text = "未接続"
483
484 'ボタン名の表示
485 btn_TestStart.Text = "試験開始"
486 btn_TestStop.Text = "準備完了"
```

```
487
488     'ボタンを無効にする
489     btn_TestStart.Enabled = False
490     btn_TestStop.Enabled = False
491     btn_RecallMemory.Enabled = False
492
493 End Sub
494
495 ''' <summary>
496 ''' VISA接続が関係するボタンの有効無効を切り替える(フォーム依存)
497 ''' </summary>
498 ''' <param name="Value"></param>
499 Private Sub ConnectButtonEnable(Value As Boolean)
500
501     btn_Connect.Enabled = Value           '接続ボタン 有効/無効
502     btn_FindResource.Enabled = Value      '検索ボタン 有効/無効
503     cmb_VisaResourceName.Enabled = Value  'VISAアドレス選択コンボボックス 有効/無効
504
505 End Sub
506
507 ''' <summary>
508 ''' ファイルパス取得処理(フォーム依存)
509 ''' </summary>
510 Private Sub GetSaveFilePath()
511
512     Dim DataFolder As String
513     Dim defaultPath As String
514     '-----
515     '試験データ保存先フォルダとして最初に提示する場所を取得
516     '-----
517     defaultPath = System.Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments)
518
519     DataFolder = file.GetFolderPath(defaultPath, "試験データ保存先のフォルダを指定してください")
520
521     If DataFolder <> "" Then           'フォルダパスの戻り値がNullでなければ
```

```
522         tb_DataSaveFolderName.Text = DataFolder
523     End If
524
525 End Sub
526
527 ''' <summary>
528 ''' メモリリコールに関する処理(フォーム依存)
529 ''' </summary>
530 Private Sub RecallMemory()
531
532     Dim iMemoryNo As Integer
533
534     '-----
535     ' メモリ番号をテキストボックスから取得する
536     '-----
537     Try
538         iMemoryNo = Convert.ToInt32(tb_MemNo.Text) 'tb_MemNoに入力された数値をint型に変換
539
540     Catch ex As Exception
541         MessageBox.Show("読み出したいメモリ番号を半角数字で入力してください",
542             "エラー",
543             MessageBoxButtons.OK,
544             MessageBoxIcon.Error)
545         Exit Sub
546     End Try
547
548     '-----
549     ' 指定したメモリを読みだし、取得したメモリ名をテキストボックスに表示する
550     '-----
551     tb_MemName.Text = Tos6200Func.GetMemName(Tos6200, iMemoryNo)
552
553     '-----
554     ' TOSをLOCAL状態に戻す
555     '-----
556     Tos6200.VisaSendCommand("LOC")
```


557

558

559 End Sub

560

561 End Class

562