```vb

Imports System.Linq.Expressions
Imports System.Net.Http.Headers
Imports System.Runtime.CompilerServices
''' <summary>
'''
''' </summary>
'''
Partial Public Class measureTask
    Public mParam As MeasureParameter
    Public cParam As ControlParameter
    ''
    Public Sub New(ByVal tasktype As String)
        mParam.TaskType = tasktype
        InitMparam()
    End Sub
    Private Sub InitMparam()
        mParam.Arraytype = 0
        mParam.NoAdChUse = 10
        mParam.MaxV = 2
        mParam.MaxI = 2
        mParam.IpMode = 0
        mParam.MaxVval = maxVvals(mParam.MaxV)
        mParam.MaxIval = maxIvals(mParam.MaxI)

        Select Case mParam.TaskType
            Case "DC" ' Or "dc" Or "Dc" Or "dC"
                mParam.CycleLen = 4
                mParam.CycleLenVal = cyclelenvalsDC(mParam.CycleLen)
                'mParam.maxN = 10
            Case "IP" ' Or "ip" Or "Ip" Or "iP"
                mParam.CycleLen = 0
                mParam.CycleLenVal = cyclelenvalsIP(mParam.CycleLen)
                ResetIPparam()
            Case "SP" 'Or "sp" Or "Sp" Or "sP"
                mParam.CycleLen = 0
                mParam.CycleLenVal = cyclelenvalsSP(mParam.CycleLen)
        End Select
        mParam.MaxN = 10
        mParam.Elspacing = 5.0
        mParam.NoStack = 1
        mParam.ErrorLimit = 5.0
        'mParam.OnOff = False
        mParam.Interval = interval
        mParam.EstTime = 0
        mParam.ElInc = 1
        ResetOnOffTime()

    End Sub
    Public Sub ResetIPparam()
        mParam.Vdly = ipParam(mParam.CycleLen, mParam.IpMode, 0)
        mParam.Mdly = ipParam(mParam.CycleLen, mParam.IpMode, 1)
        ReDim mParam.Tw(19)
        For i As Integer = 0 To 19
            mParam.Tw(i) = ipParam(mParam.CycleLen, mParam.IpMode, i + 2)
        Next
```

```
57      End Sub
58      Public Sub ResetOnOffTime()
59          If mParam.TaskType = "DC" Then
60              mParam.OnTime = dcOntime(mParam.CycleLen)
61              mParam.OffTime = dcOfftime(mParam.CycleLen)
62          ElseIf mParam.TaskType = "IP" Then
63              mParam.OnTime = ipOntime(mParam.CycleLen)
64              mParam.OffTime = ipOfftime(mParam.CycleLen)
65          End If
66      End Sub
67      Public Sub EstimateTime()
68          mParam.EstTime = mParam.CycleLenVal * mParam.NoCmd * mParam.NoStack *
              EsttimeMult
69      End Sub
70      Public Sub ResetControlParameter()
71          cParam.TaskType = mParam.TaskType
72          cParam.JSONDir = mParam.JSONDir
73          cParam.DataDir = mParam.DataDir
74          cParam.FileNameBase = mParam.FileNameBase
75          cParam.MaxN = mParam.MaxN
76          cParam.CycleLenval = mParam.CycleLenVal
77          cParam.MaxVval = mParam.MaxVval
78          cParam.MaxIval = mParam.MaxIval
79          cParam.OnTime = mParam.OnTime
80          cParam.OffTime = mParam.OffTime
81          cParam.Vdly = mParam.Vdly
82          cParam.NoStack = mParam.NoStack
83          cParam.NoCmd = mParam.NoCmd
84          cParam.Cmds = mParam.Cmds
85          cParam.NoAdChUse = mParam.NoAdChUse
86          cParam.Arraytype = mParam.Arraytype
87          cParam.ElStart = mParam.ElStart
88          cParam.ElEnd = mParam.ElEnd
89          cParam.ElInc = mParam.ElInc
90          cParam.Elspacing = mParam.Elspacing
91
92      End Sub
93      Public Sub ResetStartDate()
94          If DateDiff(DateInterval.Minute, mParam.StartDate, mParam.EndDate) > 0
              Then
95              mParam.StartDate = mParam.StartDate.AddMinutes(timeSpan)
96          End If
97      End Sub
98      'Public Sub SetFileNameBase()
99      '    mParam.FileNameBase = (mParam.TaskType & mParam.JST.ToString("yy/MM/
          dd_HH:mm"))
100     'End Sub
101     Public Sub SetFileNameBase(ByVal str As String)
102         mParam.FileNameBase = str
103     End Sub
104     Public Sub SetJST(ByVal _d As Date)
105         'Debug.WriteLine(_d.ToString("yy/MM/dd HH:mm"))
106         mParam.JST = _d
107     End Sub
108     Public Sub CopyDirStructure(ByVal _json As String, ByVal _datadir As
          String)
```

```
109         mParam.JSONDir = _json
110         mParam.DataDir = _datadir
111     End Sub
112     '=======================================
113     '  MakeCmdSP
114     '=======================================
115     Public Sub MakeCmdSP(ByRef rNo(,) As Integer, ByRef eNo(,) As Integer,
          ByVal sbNo As Integer)
116         Dim relay() As Integer
117         Dim relay2() As Integer
118         Dim nEl As Integer = 0
119 
120         ' check number of electrodes
121         For i As Integer = 0 To sbNo - 1
122             If rNo(i, 0) > 0 Then
123                 nEl = nEl + Math.Abs(rNo(i, 0) - rNo(i, 1)) + 1
124             End If
125             If rNo(i, 2) > 0 Then
126                 nEl = nEl + Math.Abs(rNo(i, 2) - rNo(i, 3)) + 1
127             End If
128         Next
129 
130         ' make relay number vector
131         ReDim relay(nEl - 1)
132         Dim indx As Integer = 0
133         Dim ellnc As Integer
134         For i As Integer = 0 To sbNo - 1
135             If rNo(i, 0) > 0 Then
136                 ellnc = 1
137                 If rNo(i, 0) > rNo(i, 1) Then
138                     ellnc = -1
139                 End If
140                 For j As Integer = rNo(i, 0) To rNo(i, 1) Step ellnc
141                     relay(indx) = j
142                     indx += 1
143                 Next
144             End If
145             If rNo(i, 2) > 0 Then
146                 ellnc = 1
147                 If rNo(i, 2) > rNo(i, 3) Then
148                     ellnc = -1
149                 End If
150                 For j As Integer = rNo(i, 2) To rNo(i, 3) Step ellnc
151                     relay(indx) = j
152                     indx += 1
153                 Next
154             End If
155         Next
156 
157         ' extract relay numbers to use measurement
158         indx = 0
159         nEl = Int((mParam.ElEnd - mParam.ElStart) / mParam.Ellnc) + 1
160         ReDim relay2(nEl - 1)
161         For i As Integer = mParam.ElStart To mParam.ElEnd Step mParam.Ellnc
162             relay2(indx) = relay(i - 1)
163             indx += 1
```

```
164         Next
165
166         '
167         ' make cmd
168         '
169         Dim ncmd1 As Integer = CInt((nEl - 1) / mParam.MaxN)
170         ReDim mParam.Cmd1(ncmd1, mParam.MaxN + 2)
171         ReDim mParam.NoCh(ncmd1)
172
173         For i As Integer = 0 To ncmd1
174             mParam.Cmd1(i, 0) = 0  ' Tx electrode as 0
175             mParam.Cmd1(i, 1) = 0
176
177             Dim chcnt As Integer = 0
178
179             Dim ind1 As Integer = i * mParam.MaxN
180             Dim ind2 As Integer = Math.Min((i + 1) * (mParam.MaxN), nEl - 1)
181             For j As Integer = ind1 To ind2
182                 mParam.Cmd1(i, j - ind1 + 2) = relay2(j)
183                 chcnt += 1
184             Next
185             mParam.NoCh(i) = chcnt - 1
186         Next
187
188         Dim ncmd As Integer = 0
189         For i As Integer = 0 To ncmd1
190             ncmd += Math.Ceiling(mParam.NoCh(i) / mParam.NoAdChUse)
191         Next
192
193         ReDim mParam.Cmds(ncmd - 1, mParam.NoAdChUse + 2)
194         'Dim indxC As Integer = 0
195         Dim indxP As Integer = 0
196         Dim indxR As Integer = 0
197         mParam.NoCmd = 0
198         For i As Integer = 0 To ncmd1
199             For j As Integer = 0 To Math.Ceiling(mParam.NoCh(i) /
                  mParam.NoAdChUse) - 1
200                 '============== debug =================
201                 'System.Console.WriteLine(mParam.nCh(i))
202                 mParam.Cmds(indxR, 0) = mParam.Cmd1(i, 0)
203                 mParam.Cmds(indxR, 1) = mParam.Cmd1(i, 1)
204
205                 For k As Integer = indxP + 2 To Math.Min(mParam.NoAdChUse +
                      indxP + 2, mParam.NoCh(i) + 2)
206                     mParam.Cmds(indxR, k - indxP) = mParam.Cmd1(i, k)
207                 Next
208                 mParam.NoCmd += 1
209                 indxP += mParam.NoAdChUse
210                 indxR += 1
211                 If indxP >= mParam.NoCh(i) Then
212                     indxP = 0
213                 End If
214             Next
215         Next
216     End Sub
217     '=====================================
```

```
218     '  MakeCmd
219     '=======================================
220     Public Sub MakeCmd(ByRef rNo(,) As Integer, ByRef eNo(,) As Integer, ByVal
        sbNo As Integer)
221         Dim relay() As Integer
222         Dim relay2() As Integer
223         Dim nEl As Integer = 0
224 
225         ' check number of electrodes
226         For i As Integer = 0 To sbNo - 1
227             If rNo(i, 0) > 0 Then
228                 nEl = nEl + Math.Abs(rNo(i, 0) - rNo(i, 1)) + 1
229             End If
230             If rNo(i, 2) > 0 Then
231                 nEl = nEl + Math.Abs(rNo(i, 2) - rNo(i, 3)) + 1
232             End If
233         Next
234 
235         ' make relay number vector
236         ReDim relay(nEl - 1)
237         Dim indx As Integer = 0
238         Dim elInc As Integer
239         For i As Integer = 0 To sbNo - 1
240             If rNo(i, 0) > 0 Then
241                 elInc = 1
242                 If rNo(i, 0) > rNo(i, 1) Then
243                     elInc = -1
244                 End If
245                 For j As Integer = rNo(i, 0) To rNo(i, 1) Step elInc
246                     relay(indx) = j
247                     indx += 1
248                 Next
249             End If
250             If rNo(i, 2) > 0 Then
251                 elInc = 1
252                 If rNo(i, 2) > rNo(i, 3) Then
253                     elInc = -1
254                 End If
255                 For j As Integer = rNo(i, 2) To rNo(i, 3) Step elInc
256                     relay(indx) = j
257                     indx += 1
258                 Next
259             End If
260         Next
261 
262         ' extract relay numbers to use measurement
263         indx = 0
264         nEl = Int((mParam.ElEnd - mParam.ElStart) / mParam.ElInc) + 1
265         ReDim relay2(nEl - 1)
266         For i As Integer = mParam.ElStart To mParam.ElEnd Step mParam.ElInc
267             relay2(indx) = relay(i - 1)
268             indx += 1
269         Next
270 
271         Select Case mParam.Arraytype
272             Case 0 ' dipole dipole
```

```
273
274                 ReDim mParam.Cmd1(nEl - 4, mParam.MaxN + 2)
275                 ReDim mParam.NoCh(nEl - 4)
276
277                 For i As Integer = 0 To nEl - 4
278                     ' System.Console.WriteLine(i)
279                     mParam.Cmd1(i, 0) = relay2(i)
280                     mParam.Cmd1(i, 1) = relay2(i + 1)
281                     Dim chcnt As Integer = 0
282                     For j As Integer = i + 2 To Math.Min(i + mParam.MaxN + 2,
                    nEl - 1) '(i+12, nEl - 1) <- this gives error
283                         mParam.Cmd1(i, j - i) = relay2(j)
284                         chcnt += 1
285                     Next
286                     mParam.NoCh(i) = chcnt - 1
287                 Next
288
289                 Dim ncmd As Integer = 0
290                 For i As Integer = 0 To nEl - 4
291                     ncmd += Math.Ceiling(mParam.NoCh(i) / mParam.NoAdChUse)
292                 Next
293
294                 '============== debug =================
295                 'System.Console.WriteLine("ncmd{0}", ncmd)
296
297                 ReDim mParam.Cmds(ncmd - 1, mParam.NoAdChUse + 2)
298                 'Dim indxC As Integer = 0
299                 Dim indxP As Integer = 0
300                 Dim indxR As Integer = 0
301                 mParam.NoCmd = 0
302                 For i As Integer = 0 To nEl - 4
303                     For j As Integer = 0 To Math.Ceiling(mParam.NoCh(i) /
                    mParam.NoAdChUse) - 1
304                         '============== debug =================
305                         'System.Console.WriteLine(mParam.nCh(i))
306                         mParam.Cmds(indxR, 0) = mParam.Cmd1(i, 0)
307                         mParam.Cmds(indxR, 1) = mParam.Cmd1(i, 1)
308
309                         For k As Integer = indxP + 2 To Math.Min
                    (mParam.NoAdChUse + indxP + 2, mParam.NoCh(i) + 2)
310                             mParam.Cmds(indxR, k - indxP) = mParam.Cmd1(i, k)
311                         Next
312                         mParam.NoCmd += 1
313                         indxP += mParam.NoAdChUse
314                         indxR += 1
315                         If indxP >= mParam.NoCh(i) Then
316                             indxP = 0
317                         End If
318                     Next
319                 Next
320                 '============== debug =================
321                 'System.Console.WriteLine("mParam.ncmd{0}", mParam.nCmd)
322
323             Case 1 ' pole pole
324
325             Case 2 ' modified pole pole
```

```
326
327             Case 3 ' Wenner
328
329             Case 4 ' schlumberger
330
331             Case 5 'modified pole pole
332
333         End Select
334
335
336     End Sub
337 End Class
338
339
```