

```
1 VERSION 5.00
2 Begin VB.Form Form1
3     BorderStyle = 1 'Fixed Single
4     Caption = "Routing Demo"
5     ClientHeight = 6615
6     ClientLeft = 45
7     ClientTop = 330
8     ClientWidth = 6615
9     LinkTopic = "Form1"
10    MaxButton = 0 'False
11    MinButton = 0 'False
12    ScaleHeight = 441
13    ScaleMode = 3 'Pixel
14    ScaleWidth = 441
15    StartUpPosition = 3 'Windows Default
16    Begin VB.CheckBox chkRough
17        Caption = "Show rough routing"
18        Height = 255
19        Left = 120
20        TabIndex = 63
21        Top = 4320
22        Width = 4095
23    End
24    Begin VB.CommandButton cmdRoute
25        Caption = "Route"
26        Height = 375
27        Left = 4320
28        TabIndex = 62
29        Top = 840
30        Width = 2175
31    End
32    Begin VB.PictureBox Picture1
33        AutoRedraw = -1 'True
34        BackColor = &H00000000&
35        BorderStyle = 0 'None
36        ForeColor = &H00FFFFFF&
37        Height = 4095
38        Left = 120
39        ScaleHeight = 273
40        ScaleMode = 3 'Pixel
41        ScaleWidth = 273
42        TabIndex = 0
43        Top = 120
44        Width = 4095
45        Begin VB.CommandButton cmdS
46            Caption = "S"
47            Height = 255
48            Index = 26
49            Left = 2640
50            TabIndex = 60
51            Top = 3120
52            Width = 255
53        End
54        Begin VB.CommandButton cmdS
55            Caption = "S"
56            Height = 255
57            Index = 25
58            Left = 1680
59            TabIndex = 59
60            Top = 3120
61            Width = 255
62        End
End
```

```
63      Begin VB.CommandButton cmdS
64          Caption      = "S"
65          Height       = 255
66          Index        = 24
67          Left         = 720
68          TabIndex     = 58
69          Top          = 3120
70          Width        = 255
71      End
72      Begin VB.CommandButton cmdS
73          Caption      = "S"
74          Height       = 255
75          Index        = 23
76          Left         = 3120
77          TabIndex     = 57
78          Top          = 2640
79          Width        = 255
80      End
81      Begin VB.CommandButton cmdS
82          Caption      = "S"
83          Height       = 255
84          Index        = 22
85          Left         = 2640
86          TabIndex     = 56
87          Top          = 2640
88          Width        = 255
89      End
90      Begin VB.CommandButton cmdS
91          Caption      = "S"
92          Height       = 255
93          Index        = 21
94          Left         = 2160
95          TabIndex     = 55
96          Top          = 2640
97          Width        = 255
98      End
99      Begin VB.CommandButton cmdS
100         Caption      = "S"
101         Height       = 255
102         Index        = 20
103         Left         = 1680
104         TabIndex     = 54
105         Top          = 2640
106         Width        = 255
107     End
108     Begin VB.CommandButton cmdN
109         Caption      = "N"
110         Height       = 255
111         Index        = 8
112         Left         = 3120
113         TabIndex     = 53
114         Top          = 3120
115         Width        = 255
116     End
117     Begin VB.CommandButton cmdS
118         Caption      = "S"
119         Height       = 255
120         Index        = 19
121         Left         = 1200
122         TabIndex     = 52
123         Top          = 2640
124         Width        = 255
```

```
125     End
126     Begin VB.CommandButton cmdN
127         Caption       = "N"
128         Height        = 255
129         Index         = 7
130         Left          = 2160
131         TabIndex      = 51
132         Top           = 3120
133         Width         = 255
134     End
135     Begin VB.CommandButton cmdS
136         Caption       = "S"
137         Height        = 255
138         Index         = 18
139         Left          = 720
140         TabIndex      = 50
141         Top           = 2640
142         Width         = 255
143     End
144     Begin VB.CommandButton cmdN
145         Caption       = "N"
146         Height        = 255
147         Index         = 6
148         Left          = 1200
149         TabIndex      = 49
150         Top           = 3120
151         Width         = 255
152     End
153     Begin VB.CommandButton cmdS
154         Caption       = "S"
155         Height        = 255
156         Index         = 17
157         Left          = 2640
158         TabIndex      = 48
159         Top           = 2160
160         Width         = 255
161     End
162     Begin VB.CommandButton cmdS
163         Caption       = "S"
164         Height        = 255
165         Index         = 16
166         Left          = 1680
167         TabIndex      = 47
168         Top           = 2160
169         Width         = 255
170     End
171     Begin VB.CommandButton cmdS
172         Caption       = "S"
173         Height        = 255
174         Index         = 15
175         Left          = 720
176         TabIndex      = 46
177         Top           = 2160
178         Width         = 255
179     End
180     Begin VB.CommandButton cmdS
181         Caption       = "S"
182         Height        = 255
183         Index         = 14
184         Left          = 3120
185         TabIndex      = 45
186         Top           = 1680
```

```
187         Width           = 255
188     End
189     Begin VB.CommandButton cmdS
190         Caption           = "S"
191         Height            = 255
192         Index             = 13
193         Left              = 2640
194         TabIndex          = 44
195         Top               = 1680
196         Width            = 255
197     End
198     Begin VB.CommandButton cmdS
199         Caption           = "S"
200         Height            = 255
201         Index             = 12
202         Left              = 2160
203         TabIndex          = 43
204         Top               = 1680
205         Width            = 255
206     End
207     Begin VB.CommandButton cmdS
208         Caption           = "S"
209         Height            = 255
210         Index             = 11
211         Left              = 1680
212         TabIndex          = 42
213         Top               = 1680
214         Width            = 255
215     End
216     Begin VB.CommandButton cmdN
217         Caption           = "N"
218         Height            = 255
219         Index             = 5
220         Left              = 3120
221         TabIndex          = 41
222         Top               = 2160
223         Width            = 255
224     End
225     Begin VB.CommandButton cmdS
226         Caption           = "S"
227         Height            = 255
228         Index             = 10
229         Left              = 1200
230         TabIndex          = 40
231         Top               = 1680
232         Width            = 255
233     End
234     Begin VB.CommandButton cmdN
235         Caption           = "N"
236         Height            = 255
237         Index             = 4
238         Left              = 2160
239         TabIndex          = 39
240         Top               = 2160
241         Width            = 255
242     End
243     Begin VB.CommandButton cmdS
244         Caption           = "S"
245         Height            = 255
246         Index             = 9
247         Left              = 720
248         TabIndex          = 38
```

```
249         Top           = 1680
250         Width          = 255
251     End
252     Begin VB.CommandButton cmdN
253         Caption         = "N"
254         Height          = 255
255         Index           = 3
256         Left            = 1200
257         TabIndex        = 37
258         Top             = 2160
259         Width           = 255
260     End
261     Begin VB.CommandButton cmdS
262         Caption         = "S"
263         Height          = 255
264         Index           = 8
265         Left            = 2640
266         TabIndex        = 36
267         Top             = 1200
268         Width           = 255
269     End
270     Begin VB.CommandButton cmdS
271         Caption         = "S"
272         Height          = 255
273         Index           = 7
274         Left            = 1680
275         TabIndex        = 35
276         Top             = 1200
277         Width           = 255
278     End
279     Begin VB.CommandButton cmdS
280         Caption         = "S"
281         Height          = 255
282         Index           = 6
283         Left            = 720
284         TabIndex        = 34
285         Top             = 1200
286         Width           = 255
287     End
288     Begin VB.CommandButton cmdS
289         Caption         = "S"
290         Height          = 255
291         Index           = 5
292         Left            = 3120
293         TabIndex        = 33
294         Top             = 720
295         Width           = 255
296     End
297     Begin VB.CommandButton cmdS
298         Caption         = "S"
299         Height          = 255
300         Index           = 4
301         Left            = 2640
302         TabIndex        = 32
303         Top             = 720
304         Width           = 255
305     End
306     Begin VB.CommandButton cmdS
307         Caption         = "S"
308         Height          = 255
309         Index           = 3
310         Left            = 2160
```

```
311         TabIndex      = 31
312         Top            = 720
313         Width         = 255
314     End
315     Begin VB.CommandButton cmdS
316         Caption       = "S"
317         Height        = 255
318         Index         = 2
319         Left          = 1680
320         TabIndex     = 30
321         Top           = 720
322         Width        = 255
323     End
324     Begin VB.CommandButton cmdS
325         Caption       = "S"
326         Height        = 255
327         Index         = 1
328         Left          = 1200
329         TabIndex     = 29
330         Top           = 720
331         Width        = 255
332     End
333     Begin VB.CommandButton cmdN
334         Caption       = "N"
335         Height        = 255
336         Index         = 2
337         Left          = 3120
338         TabIndex     = 28
339         Top           = 1200
340         Width        = 255
341     End
342     Begin VB.CommandButton cmdN
343         Caption       = "N"
344         Height        = 255
345         Index         = 1
346         Left          = 2160
347         TabIndex     = 27
348         Top           = 1200
349         Width        = 255
350     End
351     Begin VB.CommandButton cmdS
352         Caption       = "S"
353         Height        = 255
354         Index         = 0
355         Left          = 720
356         TabIndex     = 26
357         Top           = 720
358         Width        = 255
359     End
360     Begin VB.CommandButton cmdN
361         Caption       = "N"
362         Height        = 255
363         Index         = 0
364         Left          = 1200
365         TabIndex     = 25
366         Top           = 1200
367         Width        = 255
368     End
369     Begin VB.CommandButton cmdPin
370         Caption       = "24"
371         Height        = 375
372         Index         = 23
```

```
373         Left           = 3000
374         TabIndex      = 24
375         Top            = 3720
376         Width         = 495
377     End
378     Begin VB.CommandButton cmdPin
379         Caption        = "23"
380         Height         = 375
381         Index          = 22
382         Left           = 2520
383         TabIndex      = 23
384         Top            = 3720
385         Width         = 495
386     End
387     Begin VB.CommandButton cmdPin
388         Caption        = "22"
389         Height         = 375
390         Index          = 21
391         Left           = 2040
392         TabIndex      = 22
393         Top            = 3720
394         Width         = 495
395     End
396     Begin VB.CommandButton cmdPin
397         Caption        = "21"
398         Height         = 375
399         Index          = 20
400         Left           = 1560
401         TabIndex      = 21
402         Top            = 3720
403         Width         = 495
404     End
405     Begin VB.CommandButton cmdPin
406         Caption        = "20"
407         Height         = 375
408         Index          = 19
409         Left           = 1080
410         TabIndex      = 20
411         Top            = 3720
412         Width         = 495
413     End
414     Begin VB.CommandButton cmdPin
415         Caption        = "19"
416         Height         = 375
417         Index          = 18
418         Left           = 600
419         TabIndex      = 19
420         Top            = 3720
421         Width         = 495
422     End
423     Begin VB.CommandButton cmdPin
424         Caption        = "18"
425         Height         = 495
426         Index          = 17
427         Left           = 3720
428         TabIndex      = 18
429         Top            = 3000
430         Width         = 375
431     End
432     Begin VB.CommandButton cmdPin
433         Caption        = "17"
434         Height         = 495
```

```
435         Index           = 16
436         Left            = 0
437         TabIndex       = 17
438         Top             = 3000
439         Width          = 375
440     End
441     Begin VB.CommandButton cmdPin
442         Caption         = "16"
443         Height          = 495
444         Index           = 15
445         Left            = 3720
446         TabIndex       = 16
447         Top             = 2520
448         Width          = 375
449     End
450     Begin VB.CommandButton cmdPin
451         Caption         = "15"
452         Height          = 495
453         Index           = 14
454         Left            = 0
455         TabIndex       = 15
456         Top             = 2520
457         Width          = 375
458     End
459     Begin VB.CommandButton cmdPin
460         Caption         = "14"
461         Height          = 495
462         Index           = 13
463         Left            = 3720
464         TabIndex       = 14
465         Top             = 2040
466         Width          = 375
467     End
468     Begin VB.CommandButton cmdPin
469         Caption         = "13"
470         Height          = 495
471         Index           = 12
472         Left            = 0
473         TabIndex       = 13
474         Top             = 2040
475         Width          = 375
476     End
477     Begin VB.CommandButton cmdPin
478         Caption         = "12"
479         Height          = 495
480         Index           = 11
481         Left            = 3720
482         TabIndex       = 12
483         Top             = 1560
484         Width          = 375
485     End
486     Begin VB.CommandButton cmdPin
487         Caption         = "11"
488         Height          = 495
489         Index           = 10
490         Left            = 0
491         TabIndex       = 11
492         Top             = 1560
493         Width          = 375
494     End
495     Begin VB.CommandButton cmdPin
496         Caption         = "10"
```

```
497         Height           = 495
498         Index             = 9
499         Left              = 3720
500         TabIndex          = 10
501         Top               = 1080
502         Width             = 375
503     End
504     Begin VB.CommandButton cmdPin
505         Caption           = "9"
506         Height           = 495
507         Index             = 8
508         Left              = 0
509         TabIndex          = 9
510         Top               = 1080
511         Width             = 375
512     End
513     Begin VB.CommandButton cmdPin
514         Caption           = "8"
515         Height           = 495
516         Index             = 7
517         Left              = 3720
518         TabIndex          = 8
519         Top               = 600
520         Width             = 375
521     End
522     Begin VB.CommandButton cmdPin
523         Caption           = "7"
524         Height           = 495
525         Index             = 6
526         Left              = 0
527         TabIndex          = 7
528         Top               = 600
529         Width             = 375
530     End
531     Begin VB.CommandButton cmdPin
532         Caption           = "6"
533         Height           = 375
534         Index             = 5
535         Left              = 3000
536         TabIndex          = 6
537         Top               = 0
538         Width             = 495
539     End
540     Begin VB.CommandButton cmdPin
541         Caption           = "5"
542         Height           = 375
543         Index             = 4
544         Left              = 2520
545         TabIndex          = 5
546         Top               = 0
547         Width             = 495
548     End
549     Begin VB.CommandButton cmdPin
550         Caption           = "4"
551         Height           = 375
552         Index             = 3
553         Left              = 2040
554         TabIndex          = 4
555         Top               = 0
556         Width             = 495
557     End
558     Begin VB.CommandButton cmdPin
```

```
559         Caption           =   "3"
560         Height            =   375
561         Index             =   2
562         Left              =   1560
563         TabIndex          =   3
564         Top               =   0
565         Width             =   495
566     End
567     Begin VB.CommandButton cmdPin
568         Caption           =   "2"
569         Height            =   375
570         Index             =   1
571         Left              =   1080
572         TabIndex          =   2
573         Top               =   0
574         Width             =   495
575     End
576     Begin VB.CommandButton cmdPin
577         Caption           =   "1"
578         Height            =   375
579         Index             =   0
580         Left              =   600
581         TabIndex          =   1
582         Top               =   0
583         Width             =   495
584     End
585 End
586     Begin VB.Label Label1
587         Caption           =   "1. Create Netlist by clicking on pins and
nodes"
588         Height            =   495
589         Left              =   4320
590         TabIndex          =   61
591         Top               =   120
592         Width             =   2295
593     End
594 End
595 Attribute VB_Name = "Form1"
596 Attribute VB_GlobalNameSpace = False
597 Attribute VB_Creatable = False
598 Attribute VB_PredeclaredId = True
599 Attribute VB_Exposed = False
600 Dim StartEnd As Boolean
601
602 Dim StartLoc(100) As String
603 Dim EndLoc(100) As String
604
605 Dim PinNet(100) As Integer
606 Dim NNetIn(100) As Integer
607 Dim SNetIn(100) As Integer
608 Dim NNetOut(100) As Integer
609 Dim SNetOut(100) As Integer
610
611 Dim NetLen(100) As Integer
612 Dim Net(100, 100) As String
613 Dim Nets As Integer
614
615 Dim NetCost(100) As Integer
616 Dim HNetBlockade(100, 6, 7) As Boolean
617 Dim VNetBlockade(100, 7, 6) As Boolean
618
619 Dim HChannelNum(6, 7) As Integer
```

```
620 Dim VChannelNum(7, 6) As Integer
621 Dim HChannel(6, 7, 100) As Integer
622 Dim VChannel(7, 6, 100) As Integer
623
624 Dim HChannelTempB(6, 7) As Boolean
625 Dim VChannelTempB(7, 6) As Boolean
626
627 Dim TempNetX(100) As Integer
628 Dim TempNetY(100) As Integer
629 Dim TempNetH(100) As Boolean
630 Dim TempStart(100) As Boolean
631 Dim TempExpandedFrom(100) As Integer
632
633 Dim TempNets As Integer
634
635 Dim CurrentX1(100) As Integer
636 Dim CurrentY1(100) As Integer
637 Dim CurrentH1(100) As Boolean
638
639 Dim CurrentX2(100) As Integer
640 Dim CurrentY2(100) As Integer
641 Dim CurrentH2(100) As Boolean
642
643 Const FromStart = 0
644 Const TopRight = 1
645 Const BottomRight = 2
646 Const TopLeft = 3
647 Const BottomLeft = 4
648 Const FromLeft = 5
649 Const FromRight = 6
650 Const FromTop = 7
651 Const FromBottom = 8
652
653 Dim NetTree(40000) As Tree
654 Dim NetTreeLen As Long
655
656 Dim CostMin As Integer
657 Dim CostMinNode As Integer
658 Dim CostMinTopLeft As Boolean
659 Dim CostMinH As Boolean
660 Dim CostMinOneChannel As Boolean
661
662 Dim MinNetDist As Integer
663
664 Dim ToNet As Boolean
665
666 Dim RoutingLine(10000) As Lines
667 Dim RoutingLines As Long
668
669 Dim n As Integer
670
671 Private Sub cmdN_Click(Index As Integer)
672     'N = neuron output
673     'n = neuron input
674     If StartEnd Then
675         StartLoc(n) = "N" & Right("000" & Trim(Str(Index)), 3)
676     Else
677         EndLoc(n) = "n" & Right("000" & Trim(Str(Index)), 3)
678         NetLine n
679         n = n + 1
680     End If
681
```

```
682     StartEnd = Not StartEnd
683 End Sub
684
685 Private Sub cmdPin_Click(Index As Integer)
686     If StartEnd Then
687         StartLoc(n) = "P" & Right("000" & Trim(Str(Index)), 3)
688     Else
689         EndLoc(n) = "P" & Right("000" & Trim(Str(Index)), 3)
690         NetLine n
691         n = n + 1
692     End If
693
694     StartEnd = Not StartEnd
695 End Sub
696
697 Private Sub cmdRoute_Click()
698     Route
699 End Sub
700
701 Private Sub cmdS_Click(Index As Integer)
702     'S = synapse output
703     's = synapse input
704     If StartEnd Then
705         StartLoc(n) = "S" & Right("000" & Trim(Str(Index)), 3)
706     Else
707         EndLoc(n) = "s" & Right("000" & Trim(Str(Index)), 3)
708         NetLine n
709         n = n + 1
710     End If
711
712     StartEnd = Not StartEnd
713 End Sub
714
715 Private Sub NetLine(n As Integer)
716     Dim stnum As Integer, endnum As Integer
717     Dim stx As Integer, sty As Integer, endx As Integer, endy As Integer
718     Dim net1 As Integer, net2 As Integer
719
720     stnum = Val(Right(StartLoc(n), 3))
721     Select Case UCase(Left(StartLoc(n), 1))
722     Case "P"
723         With cmdPin(stnum)
724             If stnum <= 5 Then
725                 stx = .Left + .Width / 2
726                 sty = .Top + .Height
727             ElseIf stnum >= 18 Then
728                 stx = .Left + .Width / 2
729                 sty = .Top
730             ElseIf stnum Mod 2 = 0 Then
731                 stx = .Left + .Width
732                 sty = .Top + .Height / 2
733             Else
734                 stx = .Left
735                 sty = .Top + .Height / 2
736             End If
737             net1 = PinNet(stnum)
738         End With
739     Case "N"
740         With cmdN(stnum)
741             stx = .Left + .Width
742             sty = .Top + .Height
743             net1 = NNetOut(stnum)
```

```
744         End With
745     Case "S"
746         With cmdS(stnum)
747             stx = .Left + .Width
748             sty = .Top + .Height
749             net1 = SNetOut(stnum)
750         End With
751     End Select
752
753     endnum = Val(Right(EndLoc(n), 3))
754     Select Case UCase(Left(EndLoc(n), 1))
755     Case "P"
756         With cmdPin(endnum)
757             If endnum <= 5 Then
758                 endx = .Left + .Width / 2
759                 endy = .Top + .Height
760             ElseIf endnum >= 18 Then
761                 endx = .Left + .Width / 2
762                 endy = .Top
763             ElseIf endnum Mod 2 = 0 Then
764                 endx = .Left + .Width
765                 endy = .Top + .Height / 2
766             Else
767                 endx = .Left
768                 endy = .Top + .Height / 2
769             End If
770             net2 = PinNet(endnum)
771         End With
772     Case "N"
773         With cmdN(endnum)
774             endx = .Left
775             endy = .Top
776             net2 = NNetIn(endnum)
777         End With
778     Case "S"
779         With cmdS(endnum)
780             endx = .Left
781             endy = .Top
782             net2 = SNetIn(endnum)
783         End With
784     End Select
785
786     If net1 = 0 And net2 = 0 Then
787         Nets = Nets + 1
788         net1 = Nets
789         net2 = Nets
790         NetLen(Nets) = NetLen(Nets) + 1
791         Net(Nets, NetLen(Nets)) = StartLoc(n)
792         NetLen(Nets) = NetLen(Nets) + 1
793         Net(Nets, NetLen(Nets)) = EndLoc(n)
794     ElseIf net1 = 0 Then
795         net1 = net2
796         NetLen(net2) = NetLen(net2) + 1
797         Net(net2, NetLen(net2)) = StartLoc(n)
798     ElseIf net2 = 0 Then
799         net2 = net1
800         NetLen(net1) = NetLen(net1) + 1
801         Net(net1, NetLen(net1)) = EndLoc(n)
802     ElseIf net1 <> net2 Then
803         If net2 = Nets Then Nets = Nets - 1
804         For n = 1 To NetLen(net2)
805             Net(net1, NetLen(net1) + n) = Net(net2, n)
```

```
806         Next
807         For n = 1 To NetLen(net1)
808             Select Case UCase(Left(Net(net1, n), 1))
809                 Case "P"
810                     PinNet(Val(Right(Net(net1, n), 3))) = net2
811                 Case "N"
812                     NNetOut(Val(Right(Net(net1, n), 3))) = net2
813                 Case "S"
814                     SNetOut(Val(Right(Net(net1, n), 3))) = net2
815             End Select
816         Next
817         NetLen(net1) = NetLen(net1) + NetLen(net2)
818         NetLen(net2) = 0
819         net1 = net2 'arbitrary
820     End If
821
822     Select Case UCase(Left(StartLoc(n), 1))
823     Case "P"
824         PinNet(stnum) = net1
825     Case "N"
826         NNetOut(stnum) = net1
827     Case "S"
828         SNetOut(stnum) = net1
829     End Select
830
831     Select Case UCase(Left(EndLoc(n), 1))
832     Case "P"
833         PinNet(endnum) = net2
834     Case "N"
835         NNetIn(endnum) = net2
836     Case "S"
837         SNetIn(endnum) = net2
838     End Select
839
840     Picture1.Line (stx, sty)-(endx, endy), RGB(255, 255, 0)
841
842 End Sub
843
844 Private Function AddTempNet(H As Boolean, X As Integer, Y As Integer,
845     ExpandedFrom As Integer, Start As Boolean)
846     Dim exists As Boolean
847     exists = False
848
849     If (X < 1 Or Y < 1) Or (X > 6 And H) Or (X > 7 And Not H) Or (Y > 7
850 And H) Or (Y > 6 And Not H) Then
851         'this exceeds the array limits
852         AddTempNet = False
853         Exit Function
854     End If
855
856     For n = 1 To TempNets
857         If TempNetH(n) = H And TempNetX(n) = X And TempNetY(n) = Y Then
858             exists = True
859             Exit For
860         End If
861     Next
862
863     If Not exists Then
864         TempNets = TempNets + 1
865         TempNetX(TempNets) = X
866         TempNetY(TempNets) = Y
```

```
866     TempNetH(TempNets) = H
867     TempExpandedFrom(TempNets) = ExpandedFrom
868     TempStart(TempNets) = Start
869     AddTempNet = False 'point added
870
871     If H Then
872         HChannelTempB(X, Y) = True
873         If chkRough = 1 Then Picture1.Line (8 + 32 * X, 8 + 32 *
Y)-(40 + 32 * X, 8 + 32 * Y), RGB(255, 0, 0)
874         If ExpandedFrom = FromStart Then
875             If chkRough = 1 Then Picture1.Line (24 + 32 * X, 32 * Y
- 8)-(24 + 32 * X, 24 + 32 * Y), RGB(255, 255, 255)
876         End If
877     Else
878         VChannelTempB(X, Y) = True
879         If chkRough = 1 Then Picture1.Line (8 + 32 * X, 8 + 32 *
Y)-(8 + 32 * X, 40 + 32 * Y), RGB(255, 0, 0)
880         If ExpandedFrom = FromStart Then
881             If chkRough = 1 Then Picture1.Line (32 * X - 8, 24 + 32
* Y)-(24 + 32 * X, 24 + 32 * Y), RGB(255, 255, 255)
882         End If
883     End If
884
885     Else
886         If Start = TempStart(n) Then
887             AddTempNet = False 'point exists, this is a loop
888         Else
889             AddTempNet = True 'point exists, this is a connection
890         End If
891     End If
892
893 End Function
894
895 Private Function CheckTempNet(H As Boolean, X As Integer, Y As Integer,
ExpandedFrom As Integer, Start As Boolean, CheckTop As Boolean,
CheckBottom As Boolean, CheckLeft As Boolean, CheckRight As Boolean)
896     Select Case ExpandedFrom
897     Case FromStart
898         CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
899     Case TopLeft
900         If H Then
901             If CheckRight Then
902                 CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
903             Else
904                 CheckTempNet = False
905             End If
906         Else
907             If CheckBottom Then
908                 CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
909             Else
910                 CheckTempNet = False
911             End If
912         End If
913     Case FromTop
914         If CheckBottom Then
915             CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
916         Else
917             CheckTempNet = False
918         End If
919     Case TopRight
920         If H Then
921             If CheckLeft Then
```

```
922         CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
923     Else
924         CheckTempNet = False
925     End If
926 Else
927     If CheckBottom Then
928         CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
929     Else
930         CheckTempNet = False
931     End If
932 End If
933 Case FromLeft
934     If CheckRight Then
935         CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
936     End If
937 Case FromRight
938     If CheckLeft Then
939         CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
940     Else
941         CheckTempNet = False
942     End If
943 Case BottomLeft
944     If H Then
945         If CheckRight Then
946             CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
947         Else
948             CheckTempNet = False
949         End If
950     Else
951         If CheckTop Then
952             CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
953         Else
954             CheckTempNet = False
955         End If
956     End If
957 Case FromBottom
958     If CheckTop Then
959         CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
960     Else
961         CheckTempNet = False
962     End If
963 Case BottomRight
964     If H Then
965         If CheckLeft Then
966             CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
967         Else
968             CheckTempNet = False
969         End If
970     Else
971         If CheckTop Then
972             CheckTempNet = AddTempNet(H, X, Y, ExpandedFrom, Start)
973         Else
974             CheckTempNet = False
975         End If
976     End If
977 End Select
978 End Function
979
980 Private Sub Route()
981
982     Dim t As Integer
983
```

```
984     For t = 1 To Nets
985         RouteNet t
986     Next
987
988     For a = 1 To 6
989         For b = 1 To 7
990             Picture1.PSet (20 + 32 * a, 4 + 32 * b), 0
991             If HChannelNum(a, b) > 0 Then Picture1.Print HChannelNum(a,
b)
992             Picture1.PSet (4 + 32 * b, 20 + 32 * a), 0
993             If VChannelNum(b, a) > 0 Then Picture1.Print VChannelNum(b,
a)
994
995             'for each channel with > 8 traces, place a blockade on each
net in channel
996             'If HChannelNum(a, b) > 8 Then
997                 '    For n = 1 To HChannelNum(a, b)
998                     '        HNetBlockade(HChannel(a, b, n), a, b) = True
999                 '    Next
1000            'End If
1001            'If VChannelNum(b, a) > 8 Then
1002                '    For n = 1 To VChannelNum(b, a)
1003                    '        VNetBlockade(VChannel(b, a, n), b, a) = True
1004                '    Next
1005            'End If
1006        Next
1007    Next
1008 End Sub
1009
1010 Private Function RouteNet(t As Integer)
1011     'first pick a point to add to the net,
1012     'and another point if the net has not been
1013     'started
1014     Dim p1 As String, p2 As String
1015
1016     Dim n As Long
1017
1018     Dim plchxa As Integer, plchya As Integer, plchha As Boolean
1019     Dim plchxb As Integer, plchyb As Integer, plchhb As Boolean
1020
1021     Dim p2chxa As Integer, p2chya As Integer, p2chha As Boolean
1022     Dim p2chxb As Integer, p2chyb As Integer, p2chhb As Boolean
1023
1024     Dim p2ToLeft As Boolean, p2ToRight As Boolean, p2ToTop As Boolean,
p2ToBottom As Boolean
1025
1026     Dim connmade As Boolean, checksome As Boolean
1027
1028     Dim counter As Integer, CheckTop As Boolean, CheckBottom As Boolean,
CheckLeft As Boolean, CheckRight As Boolean
1029
1030     Dim destXa1 As Integer, destYa1 As Integer
1031     Dim destXa2 As Integer, destYa2 As Integer
1032     Dim destXb1 As Integer, destYb1 As Integer
1033     Dim destXb2 As Integer, destYb2 As Integer
1034
1035     Dim X1 As Integer, Y1 As Integer, X2 As Integer, Y2 As Integer
1036
1037     Dim node As Long
1038
1039     Picture1.Cls
1040
```

```
1041     NetCost(t) = 0
1042
1043     p1 = Net(t, 1)
1044     'find channels of start
1045     Select Case UCase(Left(p1, 1))
1046     Case "P"
1047         Select Case Val(Right(p1, 3))
1048         Case 0 To 5
1049             plchha = True
1050             plchxa = Val(Right(p1, 3)) + 1
1051             plchya = 1
1052         Case 6, 8, 10, 12, 14, 16
1053             plchha = False
1054             plchxa = 1
1055             plchya = Val(Right(p1, 3)) / 2 - 2
1056         Case 7, 9, 11, 13, 15, 17
1057             plchha = False
1058             plchxa = 7
1059             plchya = (Val(Right(p1, 3)) - 1) / 2 - 2
1060         Case 18 To 23
1061             plchha = True
1062             plchxa = Val(Right(p1, 3)) - 17
1063             plchya = 7
1064         End Select
1065     Case "N"
1066         Select Case Val(Right(p1, 3))
1067         Case 0 To 2
1068             plchha = False
1069             plchxa = Val(Right(p1, 3)) * 2 + 3
1070             plchya = 2
1071             plchhb = True
1072             plchxb = Val(Right(p1, 3)) * 2 + 2
1073             plchyb = 3
1074         Case 3 To 5
1075             plchha = False
1076             plchxa = Val(Right(p1, 3)) * 2 - 3
1077             plchya = 4
1078             plchhb = True
1079             plchxb = Val(Right(p1, 3)) * 2 - 4
1080             plchyb = 5
1081         Case 6 To 8
1082             plchha = False
1083             plchxa = Val(Right(p1, 3)) * 2 - 9
1084             plchya = 6
1085             plchhb = True
1086             plchxb = Val(Right(p1, 3)) * 2 - 10
1087             plchyb = 7
1088         End Select
1089         If Left(p1, 1) = "n" Then
1090             plchxa = plchxa - 1
1091             plchyb = plchyb - 1
1092         End If
1093     Case "S"
1094         Select Case Val(Right(p1, 3))
1095         Case 0 To 5
1096             plchha = False
1097             plchxa = Val(Right(p1, 3)) + 2
1098             plchya = 1
1099             plchhb = True
1100             plchxb = Val(Right(p1, 3)) + 1
1101             plchyb = 2
1102         Case 6 To 8
```

```
1103         plchha = False
1104         plchxa = (Val(Right(p1, 3)) - 6) * 2 + 2
1105         plchya = 2
1106         plchhb = True
1107         plchxb = (Val(Right(p1, 3)) - 6) * 2 + 1
1108         plchyb = 3
1109     Case 9 To 14
1110         plchha = False
1111         plchxa = (Val(Right(p1, 3)) - 9) + 2
1112         plchya = 3
1113         plchhb = True
1114         plchxb = (Val(Right(p1, 3)) - 9) + 1
1115         plchyb = 4
1116     Case 15 To 17
1117         plchha = False
1118         plchxa = (Val(Right(p1, 3)) - 15) * 2 + 2
1119         plchya = 4
1120         plchhb = True
1121         plchxb = (Val(Right(p1, 3)) - 15) * 2 + 1
1122         plchyb = 5
1123     Case 18 To 23
1124         plchha = False
1125         plchxa = (Val(Right(p1, 3)) - 18) + 2
1126         plchya = 5
1127         plchhb = True
1128         plchxb = (Val(Right(p1, 3)) - 18) + 1
1129         plchyb = 6
1130     Case 24 To 26
1131         plchha = False
1132         plchxa = (Val(Right(p1, 3)) - 24) * 2 + 2
1133         plchya = 6
1134         plchhb = True
1135         plchxb = (Val(Right(p1, 3)) - 24) * 2 + 1
1136         plchyb = 7
1137     End Select
1138     If Left(p1, 1) = "s" Then
1139         plchxa = plchxa - 1
1140         plchyb = plchyb - 1
1141     End If
1142 End Select
1143
1144 For p = 2 To NetLen(t)
1145     p2 = Net(t, p)
1146     'now extend both of these locations
1147
1148     'find channels of end
1149     Select Case UCase(Left(p2, 1))
1150     Case "P"
1151         Select Case Val(Right(p2, 3))
1152         Case 0 To 5
1153             p2chha = True
1154             p2chxa = Val(Right(p2, 3)) + 1
1155             p2chya = 1
1156         Case 6, 8, 10, 12, 14, 16
1157             p2chha = False
1158             p2chxa = 1
1159             p2chya = Val(Right(p2, 3)) / 2 - 2
1160         Case 7, 9, 11, 13, 15, 17
1161             p2chha = False
1162             p2chxa = 7
1163             p2chya = (Val(Right(p2, 3)) - 1) / 2 - 2
1164         Case 18 To 23
```

```
1165         p2chha = True
1166         p2chxa = Val(Right(p2, 3)) - 17
1167         p2chya = 7
1168     End Select
1169     Case "N"
1170         Select Case Val(Right(p2, 3))
1171         Case 0 To 2
1172             p2chha = False
1173             p2chxa = Val(Right(p2, 3)) * 2 + 3
1174             p2chya = 2
1175             p2chhb = True
1176             p2chxb = Val(Right(p2, 3)) * 2 + 2
1177             p2chyb = 3
1178         Case 3 To 5
1179             p2chha = False
1180             p2chxa = Val(Right(p2, 3)) * 2 - 3
1181             p2chya = 4
1182             p2chhb = True
1183             p2chxb = Val(Right(p2, 3)) * 2 - 4
1184             p2chyb = 5
1185         Case 6 To 8
1186             p2chha = False
1187             p2chxa = Val(Right(p2, 3)) * 2 - 9
1188             p2chya = 6
1189             p2chhb = True
1190             p2chxb = Val(Right(p2, 3)) * 2 - 10
1191             p2chyb = 7
1192         End Select
1193         If Left(p2, 1) = "n" Then
1194             p2chxa = p2chxa - 1
1195             p2chyb = p2chyb - 1
1196         End If
1197     Case "S"
1198         Select Case Val(Right(p2, 3))
1199         Case 0 To 5
1200             p2chha = False
1201             p2chxa = Val(Right(p2, 3)) + 2
1202             p2chya = 1
1203             p2chhb = True
1204             p2chxb = Val(Right(p2, 3)) + 1
1205             p2chyb = 2
1206         Case 6 To 8
1207             p2chha = False
1208             p2chxa = (Val(Right(p2, 3)) - 6) * 2 + 2
1209             p2chya = 2
1210             p2chhb = True
1211             p2chxb = (Val(Right(p2, 3)) - 6) * 2 + 1
1212             p2chyb = 3
1213         Case 9 To 14
1214             p2chha = False
1215             p2chxa = (Val(Right(p2, 3)) - 9) + 2
1216             p2chya = 3
1217             p2chhb = True
1218             p2chxb = (Val(Right(p2, 3)) - 9) + 1
1219             p2chyb = 4
1220         Case 15 To 17
1221             p2chha = False
1222             p2chxa = (Val(Right(p2, 3)) - 15) * 2 + 2
1223             p2chya = 4
1224             p2chhb = True
1225             p2chxb = (Val(Right(p2, 3)) - 15) * 2 + 1
1226             p2chyb = 5
```

```
1227         Case 18 To 23
1228             p2chha = False
1229             p2chxa = (Val(Right(p2, 3)) - 18) + 2
1230             p2chya = 5
1231             p2chhb = True
1232             p2chxb = (Val(Right(p2, 3)) - 18) + 1
1233             p2chyb = 6
1234         Case 24 To 26
1235             p2chha = False
1236             p2chxa = (Val(Right(p2, 3)) - 24) * 2 + 2
1237             p2chya = 6
1238             p2chhb = True
1239             p2chxb = (Val(Right(p2, 3)) - 24) * 2 + 1
1240             p2chyb = 7
1241     End Select
1242     If Left(p2, 1) = "s" Then
1243         p2chxa = p2chxa - 1
1244         p2chyb = p2chyb - 1
1245     End If
1246 End Select
1247
1248 'IMPROVEMENT: make p2 connect to any point on the net instead of
p1
1249 'search channels for net t
1250
1251 '***** Better yet, check on-the-fly if the net has been reached
1252
1253 dist = 9999
1254 MinNetDist = (p1chxa - p2chxa) ^ 2 + (p1chya - p2chya) ^ 2
1255 ToNet = False
1256
1257 For a = 1 To 6
1258     For b = 1 To 7
1259         For q = 1 To HChannelNum(a, b)
1260             If HChannel(a, b, q) = t Then
1261                 dist = (a - p2chxa) ^ 2 + (b - p2chya) ^ 2
1262                 If dist < MinNetDist Then
1263                     MinNetDist = dist
1264                     p1chha = True
1265                     p1chxa = a
1266                     p1chya = b
1267                     p1 = "P000"
1268                     p1chhb = False
1269                     p1chxb = 0
1270                     p1chyb = 0
1271                     ToNet = True
1272                 End If
1273             End If
1274         Next
1275     For q = 1 To VChannelNum(b, a)
1276         If VChannel(b, a, q) = t Then
1277             dist = (b - p2chxa) ^ 2 + (a - p2chya) ^ 2
1278             If dist < MinNetDist Then
1279                 MinNetDist = dist
1280                 p1chha = False
1281                 p1chxa = b
1282                 p1chya = a
1283                 p1 = "P000"
1284                 p1chhb = False
1285                 p1chxb = 0
1286                 p1chyb = 0
1287                 ToNet = True
```

```
1288             End If
1289         End If
1290     Next
1291 Next
1292 Next
1293
1294     'MsgBox p1 & " to " & p2 & " (net " & t & ")"
1295
1296     'reset temporary arrays
1297
1298     TempNets = 0
1299
1300     For r = 0 To 6
1301         For q = 0 To 7
1302             HChannelTempB(r, q) = False
1303             VChannelTempB(q, r) = False
1304         Next
1305     Next
1306
1307     'find relative position
1308     p2ToRight = p2chxa > p1chxa
1309     p2ToLeft = p2chxa < p1chxa
1310     p2ToBottom = p2chya > p1chya
1311     p2ToTop = p2chya < p1chya
1312
1313     If chkRough = 1 Then Picture1.Cls
1314
1315     AddTempNet p1chha, p1chxa, p1chya, 0, True 'point1 = true,
point2 = false
1316
1317     If UCase(Left(p1, 1)) = "N" Or UCase(Left(p1, 1)) = "S" Then
1318         AddTempNet p1chhb, p1chxb, p1chyb, 0, True
1319     End If
1320
1321     AddTempNet p2chha, p2chxa, p2chya, 0, False
1322
1323     If UCase(Left(p2, 1)) = "N" Or UCase(Left(p2, 1)) = "S" Then
1324         AddTempNet p2chhb, p2chxb, p2chyb, 0, False
1325     End If
1326
1327     connmade = False
1328
1329     'this step will limit the search range for the best solution
1330
1331     r = 1
1332
1333     NetTreeLen = 0
1334
1335     'special case for well placed cell: might share channel with
destination
1336     If p1chha = p2chha And p1chxa = p2chxa And p1chya = p2chya Then
1337         connmade = True
1338         AddTempNet p1chha, p1chxa, p1chya, FromStart, True
1339     ElseIf Left(p2, 1) <> "P" And p1chha = p2chhb And p1chxa =
p2chxb And p1chya = p2chyb Then
1340         connmade = True
1341         AddTempNet p1chha, p1chxa, p1chya, FromStart, True
1342     ElseIf Left(p1, 1) <> "P" And p1chhb = p2chha And p1chxb =
p2chxa And p1chyb = p2chya Then
1343         connmade = True
1344         AddTempNet p1chhb, p1chxb, p1chyb, FromStart, True
1345     ElseIf Left(p1, 1) <> "P" And Left(p2, 1) <> "P" And p1chhb =
```

```
    p2chhb And plchxb = p2chxb And plchyb = p2chyb Then
1346        connmade = True
1347        AddTempNet plchhb, plchxb, plchyb, FromStart, True
1348    End If
1349
1350    Do Until connmade
1351
1352        'counter = counter + 1
1353
1354        'If r > TempNets Then 'counter >= 3 Then
1355        '    counter = 0
1356        '    CheckTop = True
1357        '    CheckLeft = True
1358        '    CheckBottom = True
1359        '    CheckRight = True
1360        '    checksome = False
1361        '    r = 0
1362        'Else
1363        '    checksome = True
1364        'End If
1365
1366    q = TempNets
1367    For o = r To q
1368        'expand nets
1369
1370        'expand in all directions if possible
1371
1372        If checksome Then
1373            If TempStart(o) Then
1374                CheckTop = p2ToTop
1375                CheckLeft = p2ToLeft
1376                CheckBottom = p2ToBottom
1377                CheckRight = p2ToRight
1378            Else
1379                CheckTop = p2ToBottom
1380                CheckLeft = p2ToRight
1381                CheckBottom = p2ToTop
1382                CheckRight = p2ToLeft
1383            End If
1384        End If
1385
1386        If TempNetH(o) Then
1387            'horizontal channel
1388            'expand to vertical channels (X,Y) (X,Y-1) (X+1,Y)
1389            (X+1,Y-1)
1390            'and horizontal channels (X-1,Y) (X+1,Y)
1391            'try to expand mostly in the direction(s) of the
1392            target
1393            '(not yet implemented)
1394
1395            If CheckTempNet(False, TempNetX(o), TempNetY(o),
1396            TopRight, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight) _
1397            Or CheckTempNet(False, TempNetX(o), TempNetY(o) - 1,
1398            BottomRight, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight)
1399            -
1400            Or CheckTempNet(False, TempNetX(o) + 1, TempNetY(o),
1401            TopLeft, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight) _
1402            Or CheckTempNet(False, TempNetX(o) + 1, TempNetY(o)
1403            - 1, BottomLeft, TempStart(o), CheckTop, CheckBottom, CheckLeft,
1404            CheckRight) -
1405            Or CheckTempNet(True, TempNetX(o) - 1, TempNetY(o),
```

```
1399 FromRight, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight) _  
      Or CheckTempNet(True, TempNetX(o) + 1, TempNetY(o), _  
FromLeft, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight)  
Then  
1400         'connection has been made in this expansion  
1401         connmade = True  
1402     End If  
1403     Else  
1404         'vertical channel  
1405         'expand to vertical channels (X,Y-1) (X,Y+1)  
1406         'expand to horizontal channels (X-1,Y) (X,Y)  
      (X-1,Y+1) (X,Y+1)  
1407  
1408         'try to expand mostly in the direction(s) of the  
target  
1409         '(not yet implemented)  
1410  
1411         If CheckTempNet(False, TempNetX(o), TempNetY(o) - 1,  
FromBottom, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight)  
1412 -         Or CheckTempNet(False, TempNetX(o), TempNetY(o) + 1,  
FromTop, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight) _  
1413 Or CheckTempNet(True, TempNetX(o) - 1, TempNetY(o), _  
BottomRight, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight)  
1414 -         Or CheckTempNet(True, TempNetX(o), TempNetY(o),  
BottomLeft, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight)  
1415 -         Or CheckTempNet(True, TempNetX(o) - 1, TempNetY(o) +  
1, TopRight, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight)  
1416 -         Or CheckTempNet(True, TempNetX(o), TempNetY(o) + 1,  
TopLeft, TempStart(o), CheckTop, CheckBottom, CheckLeft, CheckRight)  
Then  
1417         'connection has been made in this expansion  
1418         connmade = True  
1419     End If  
1420 End If  
1421 Next  
1422  
1423     'MsgBox "working..."  
1424  
1425     r = q + 1  
1426     If r > TempNets Then 'And counter = 0 Then  
1427         'MsgBox "Cannot Route"  
1428         Exit Do  
1429     End If  
1430  
1431  
1432     Loop  
1433  
1434     'MsgBox "Initial Possibilities Found"  
1435     Picture1.Refresh  
1436  
1437     NetTreeLen = 0  
1438  
1439     CostMin = 9999  
1440  
1441     'now make trees to determine the lowest cost pathway  
1442  
1443     '           [1]  
1444     ' [2]           [3]           [4]
```

```
1445      '[5][6][7] [8][9][10] [11][12][13]
1446      '....
1447
1448      'convert this h or v channel to 2 possible switchboxes
1449      'H: (x, y) (x+1, y)
1450      'V: (x, y) (x, y+1)
1451      If p2chha Then 'H
1452          AddNetTreeNode 0, p2chxa, p2chya, FromRight, 1
1453          AddNetTreeNode 0, p2chxa + 1, p2chya, FromLeft, 1
1454      Else 'V
1455          AddNetTreeNode 0, p2chxa, p2chya, FromBottom, 1
1456          AddNetTreeNode 0, p2chxa, p2chya + 1, FromTop, 1
1457      End If
1458      'cost = 1 for first channel connections
1459
1460      If Left(p2, 1) <> "P" Then
1461          'neuron or synapse, with 2 start locations
1462          If p2chhb Then 'H
1463              AddNetTreeNode 0, p2chxb, p2chyb, FromRight, 1
1464              AddNetTreeNode 0, p2chxb + 1, p2chyb, FromLeft, 1
1465          Else 'V
1466              AddNetTreeNode 0, p2chxb, p2chyb, FromBottom, 1
1467              AddNetTreeNode 0, p2chxb, p2chyb + 1, FromTop, 1
1468          End If
1469      End If
1470
1471      'calculate switchboxes for connecting to the final destinations
1472
1473      If plchha Then 'H
1474          destXa1 = plchxa
1475          destYa1 = plchya
1476          destXa2 = plchxa + 1
1477          destYa2 = plchya
1478      Else 'V
1479          destXa1 = plchxa
1480          destYa1 = plchya
1481          destXa2 = plchxa
1482          destYa2 = plchya + 1
1483      End If
1484      'cost = 1 for first channel connections
1485
1486      If Left(p1, 1) <> "P" Then
1487          'neuron or synapse, with 2 start locations
1488          If plchhb Then 'H
1489              destXb1 = plchxb
1490              destYb1 = plchyb
1491              destXb2 = plchxb + 1
1492              destYb2 = plchyb
1493          Else 'V
1494              destXb1 = plchxb
1495              destYb1 = plchyb
1496              destXb2 = plchxb
1497              destYb2 = plchyb + 1
1498          End If
1499      End If
1500
1501      n = 1
1502
1503      CostMinOneChannel = False
1504
1505      'special case for well placed cell: might share channel with
destination
```

```
1506         If plchha = p2chha And plchxa = p2chxa And plchya = p2chya Then
1507             AddNetTreeNode 1, plchxa, plchya, FromStart, 2
1508             CostMin = 2
1509             CostMinNode = NetTreeLen
1510             CostMinTopLeft = False
1511             CostMinOneChannel = True
1512             CostMinH = plchha
1513             n = NetTreeLen + 1
1514         ElseIf Left(p2, 1) <> "P" And plchha = p2chhb And plchxa =
p2chxb And plchya = p2chyb Then
1515             AddNetTreeNode 1, plchxa, plchya, FromStart, 2
1516             CostMin = 2
1517             CostMinNode = NetTreeLen
1518             CostMinTopLeft = False
1519             CostMinOneChannel = True
1520             CostMinH = plchha
1521             n = NetTreeLen + 1
1522         ElseIf Left(p1, 1) <> "P" And plchhb = p2chha And plchxb =
p2chxa And plchyb = p2chya Then
1523             AddNetTreeNode 3, plchxb, plchyb, FromStart, 2
1524             CostMin = 2
1525             CostMinNode = NetTreeLen
1526             CostMinTopLeft = False
1527             CostMinOneChannel = True
1528             CostMinH = plchhb
1529             n = NetTreeLen + 1
1530         ElseIf Left(p1, 1) <> "P" And Left(p2, 1) <> "P" And plchhb =
p2chhb And plchxb = p2chxb And plchyb = p2chyb Then
1531             AddNetTreeNode 3, plchxb, plchyb, FromStart, 2
1532             CostMin = 2
1533             CostMinNode = NetTreeLen
1534             CostMinTopLeft = False
1535             CostMinOneChannel = True
1536             CostMinH = plchhb
1537             n = NetTreeLen + 1
1538         End If
1539
1540         Do Until n > NetTreeLen Or n > 10000
1541             'do not make loops with parent connections
1542             'also, check all connections against the boolean temp array
1543             'try to make connections in (x-1,y) (x+1,y) (x,y-1) (x,y+1)
1544
1545             node = n
1546
1547             CheckTop = True
1548             CheckBottom = True
1549             CheckLeft = True
1550             CheckRight = True
1551
1552             'MsgBox "Node " & n & ": (" & NetTree(n).SwitchBoxX & ", " &
NetTree(n).SwitchBoxY & ")" & vbCrLf & "Parent " & NetTree(n).Parent &
": (" & NetTree(NetTree(n).Parent).SwitchBoxX & ", " &
NetTree(NetTree(n).Parent).SwitchBoxY & ")"
1553
1554             'this will check if we have reached a destination switchbox
1555             'calculate cost to reach a destination channel/unit
1556             'there may be 2 costs and 2 destinations in some cases
1557
1558             With NetTree(n)
1559                 If (.SwitchBoxX = destXal And .SwitchBoxY = destYal)
1560
Then
```

```
1561         CheckTop = False
1562         CheckBottom = False
1563         CheckLeft = False
1564         CheckRight = False
1565
1566         If plchha Then 'H
1567             'we are to the left of the destination channel
1568             (X,Y)
1569             If CheckHChannelTempB(.SwitchBoxX, .SwitchBoxY)
1570                 Then
1571                     'yes, we can make the connection to the
1572                     right
1573                     If .ExtendedFrom = FromLeft Then
1574                         'cost = 2 + 1
1575                         .FinalCostA = .Cost + 3
1576                     Else
1577                         'cost = 3 + 1
1578                         .FinalCostA = .Cost + 4
1579                     End If
1580                 End If
1581             Else 'V
1582                 'we are to the top of the destination channel
1583                 (X,Y)
1584                 If CheckVChannelTempB(.SwitchBoxX, .SwitchBoxY)
1585                     Then
1586                         'yes, we can make the connection to the
1587                         bottom
1588                         If .ExtendedFrom = FromTop Then
1589                             'cost = 2 + 1
1590                             .FinalCostA = .Cost + 3
1591                         Else
1592                             'cost = 3 + 1
1593                             .FinalCostA = .Cost + 4
1594                         End If
1595                     End If
1596                 End If
1597             If .FinalCostA < CostMin Then
1598                 CostMin = .FinalCostA
1599                 CostMinNode = n
1600                 CostMinTopLeft = True
1601                 CostMinH = plchha
1602             End If
1603         ElseIf (.SwitchBoxX = destXa2 And .SwitchBoxY = destYa2)
1604             Then
1605                 CheckTop = False
1606                 CheckBottom = False
1607                 CheckLeft = False
1608                 CheckRight = False
1609
1610                 If plchha Then 'H
1611                     'we are to the right of the destination channel
1612                     (X-1,Y)
1613                     If CheckHChannelTempB(.SwitchBoxX - 1,
1614                         .SwitchBoxY) Then
1615                         'yes, we can make the connection to the left
1616                         If .ExtendedFrom = FromRight Then
1617                             'cost = 2 + 1
1618                             .FinalCostA = .Cost + 3
1619                         Else
1620                             'cost = 3 + 1
```

```
1614             .FinalCostA = .Cost + 4
1615             End If
1616         End If
1617     Else 'V
1618         'we are to the bottom of the destination channel
1619     (X,Y-1)
1620         If CheckVChannelTempB(.SwitchBoxX, .SwitchBoxY -
1621     1) Then
1622             'yes, we can make the connection to the top
1623             If .ExtendedFrom = FromBottom Then
1624                 'cost = 2 + 1
1625                 .FinalCostA = .Cost + 3
1626             Else
1627                 'cost = 3 + 1
1628                 .FinalCostA = .Cost + 4
1629             End If
1630         End If
1631     If .FinalCostA < CostMin Then
1632         CostMin = .FinalCostA
1633         CostMinNode = n
1634         CostMinTopLeft = False
1635         CostMinH = plchha
1636     End If
1637 End If
1638 If Left(p2, 1) <> "P" Then
1639     If (.SwitchBoxX = destXb1 And .SwitchBoxY = destYb1)
1640 Then
1641     CheckTop = False
1642     CheckBottom = False
1643     CheckLeft = False
1644     CheckRight = False
1645
1646     If plchhb Then 'H
1647         'we are to the left of the destination
1648     channel (X,Y)
1649         If CheckHChannelTempB(.SwitchBoxX,
1650     .SwitchBoxY) Then
1651             'yes, we can make the connection to the
1652     right
1653             If .ExtendedFrom = FromLeft Then
1654                 'cost = 2 + 1
1655                 .FinalCostB = .Cost + 3
1656             Else
1657                 'cost = 3 + 1
1658                 .FinalCostB = .Cost + 4
1659             End If
1660         End If
1661     Else 'V
1662         'we are to the top of the destination
1663     channel (X,Y)
1664         If CheckVChannelTempB(.SwitchBoxX,
1665     .SwitchBoxY) Then
1666             'yes, we can make the connection to the
1667     bottom
1668             If .ExtendedFrom = FromTop Then
1669                 'cost = 2 + 1
1670                 .FinalCostB = .Cost + 3
1671             Else
1672                 'cost = 3 + 1
```

```
1667             .FinalCostB = .Cost + 4
1668             End If
1669         End If
1670     End If
1671     If .FinalCostB < CostMin Then
1672         CostMin = .FinalCostB
1673         CostMinNode = n
1674         CostMinTopLeft = True
1675         CostMinH = plchhb
1676     End If
1677
1678     ElseIf (.SwitchBoxX = destXb2 And .SwitchBoxY =
1679 destYb2) Then
1680         CheckTop = False
1681         CheckBottom = False
1682         CheckLeft = False
1683         CheckRight = False
1684
1685         If plchhb Then 'H
1686             'we are to the right of the destination
1687             channel (X-1,Y)
1688             If CheckHChannelTempB(.SwitchBoxX - 1,
1689 .SwitchBoxY) Then
1690                 'yes, we can make the connection to the
1691                 left
1692                 If .ExtendedFrom = FromRight Then
1693                     'cost = 2 + 1
1694                     .FinalCostB = .Cost + 3
1695                 Else
1696                     'cost = 3 + 1
1697                     .FinalCostB = .Cost + 4
1698                 End If
1699             End If
1700             Else 'V
1701                 'we are to the bottom of the destination
1702                 channel (X,Y-1)
1703                 If CheckVChannelTempB(.SwitchBoxX,
1704 .SwitchBoxY - 1) Then
1705                     'yes, we can make the connection to the
1706                     top
1707                     If .ExtendedFrom = FromBottom Then
1708                         'cost = 2 + 1
1709                         .FinalCostB = .Cost + 3
1710                     Else
1711                         'cost = 3 + 1
1712                         .FinalCostB = .Cost + 4
1713                     End If
1714                 End If
1715             End If
1716         End If
1717     If .FinalCostB < CostMin Then
1718         CostMin = .FinalCostB
1719         CostMinNode = n
1720         CostMinTopLeft = False
1721         CostMinH = plchhb
1722     End If
1723 End With
1724 Do Until NetTree(node).Parent = 0
1725     node = NetTree(node).Parent
```

```
1722         'check (x-1,y)
1723         If (NetTree(node).SwitchBoxX = NetTree(n).SwitchBoxX -
1724     1) And (NetTree(node).SwitchBoxY = NetTree(n).SwitchBoxY) Then
1725             CheckLeft = False
1726         'check (x+1,y)
1727         ElseIf (NetTree(node).SwitchBoxX = NetTree(n).SwitchBoxX
1728     + 1) And (NetTree(node).SwitchBoxY = NetTree(n).SwitchBoxY) Then
1729             CheckRight = False
1730         'check (x,y-1)
1731         ElseIf (NetTree(node).SwitchBoxX =
1732     NetTree(n).SwitchBoxX) And (NetTree(node).SwitchBoxY =
1733     NetTree(n).SwitchBoxY - 1) Then
1734             CheckTop = False
1735         'check (x,y+1)
1736         ElseIf (NetTree(node).SwitchBoxX =
1737     NetTree(n).SwitchBoxX) And (NetTree(node).SwitchBoxY =
1738     NetTree(n).SwitchBoxY + 1) Then
1739             CheckBottom = False
1740         End If
1741     Loop
1742     With NetTree(n)
1743         If Not CheckVChannelTempB(.SwitchBoxX, .SwitchBoxY - 1)
1744     Then
1745             'up pathway blocked
1746             CheckTop = False
1747         End If
1748         If Not CheckVChannelTempB(.SwitchBoxX, .SwitchBoxY) Then
1749             'down pathway blocked
1750             CheckBottom = False
1751         End If
1752         If Not CheckHChannelTempB(.SwitchBoxX - 1, .SwitchBoxY)
1753     Then
1754             'left pathway blocked
1755             CheckLeft = False
1756         End If
1757         If Not CheckHChannelTempB(.SwitchBoxX, .SwitchBoxY) Then
1758             'right pathway blocked
1759             CheckRight = False
1760         End If
1761         If .ExtendedFrom = FromLeft Then
1762             CheckLeft = False
1763         ElseIf .ExtendedFrom = FromRight Then
1764             CheckRight = False
1765         ElseIf .ExtendedFrom = FromTop Then
1766             CheckTop = False
1767         ElseIf .ExtendedFrom = FromBottom Then
1768             CheckBottom = False
1769         End If
1770         If CheckLeft Then
1771             If .ExtendedFrom = FromRight Then
1772                 'cost = 2
1773                 AddNetTreeNode n, .SwitchBoxX - 1, .SwitchBoxY,
1774     FromRight, .Cost + 2
1775             Else
1776                 'cost = 3
1777                 AddNetTreeNode n, .SwitchBoxX - 1, .SwitchBoxY,
1778     FromRight, .Cost + 3
1779             End If
```

```
1774             End If
1775             If CheckRight Then
1776                 If .ExtendedFrom = FromLeft Then
1777                     'cost = 2
1778                     AddNetTreeNode n, .SwitchBoxX + 1, .SwitchBoxY,
FromLeft, .Cost + 2
1779                 Else
1780                     'cost = 3
1781                     AddNetTreeNode n, .SwitchBoxX + 1, .SwitchBoxY,
FromLeft, .Cost + 3
1782                 End If
1783             End If
1784             If CheckTop Then
1785                 If .ExtendedFrom = FromBottom Then
1786                     'cost = 2
1787                     AddNetTreeNode n, .SwitchBoxX, .SwitchBoxY - 1,
FromBottom, .Cost + 2
1788                 Else
1789                     'cost = 3
1790                     AddNetTreeNode n, .SwitchBoxX, .SwitchBoxY - 1,
FromBottom, .Cost + 3
1791                 End If
1792             End If
1793             If CheckBottom Then
1794                 If .ExtendedFrom = FromTop Then
1795                     'cost = 2
1796                     AddNetTreeNode n, .SwitchBoxX, .SwitchBoxY + 1,
FromTop, .Cost + 2
1797                 Else
1798                     'cost = 3
1799                     AddNetTreeNode n, .SwitchBoxX, .SwitchBoxY + 1,
FromTop, .Cost + 3
1800                 End If
1801             End If
1802         End With
1803
1804         n = n + 1
1805     Loop
1806
1807     'MsgBox "Min Cost: " & CostMin
1808
1809     NetCost(t) = NetCost(t) + CostMin
1810
1811     node = CostMinNode
1812
1813     If chkRough = 1 Then Picture1.Cls
1814
1815     'MsgBox "(" & NetTree(node).SwitchBoxX & "," &
NetTree(node).SwitchBoxY & ")"
1816     X1 = NetTree(node).SwitchBoxX
1817     Y1 = NetTree(node).SwitchBoxY
1818
1819     If CostMinOneChannel And Not ToNet Then
1820         If CostMinH Then
1821             Picture1.Line (24 + 32 * X1, 32 * Y1 - 8)-(24 + 32 * X1,
24 + 32 * Y1), RGB(0, 0, 255)
1822             AddToChannels X1, Y1, X1 + 1, Y1, t
1823         Else
1824             Picture1.Line (32 * X1 - 8, 24 + 32 * Y1)-(24 + 32 * X1,
24 + 32 * Y1), RGB(0, 0, 255)
1825             AddToChannels X1, Y1, X1, Y1 + 1, t
1826         End If
```

```
1827         Else
1828
1829             If CostMinH Then
1830                 If plchha Then
1831                     AddToChannels plchxa, Y1, plchxa + 1, Y1, t
1832                 Else
1833                     AddToChannels plchxb, Y1, plchxb + 1, Y1, t
1834                 End If
1835             Else
1836                 If Not plchha Then
1837                     AddToChannels X1, plchya, X1, plchya + 1, t
1838                 Else
1839                     AddToChannels X1, plchyb, X1, plchyb + 1, t
1840                 End If
1841             End If
1842
1843             Do Until NetTree(node).Parent = 0
1844                 node = NetTree(node).Parent
1845                 X2 = NetTree(node).SwitchBoxX
1846                 Y2 = NetTree(node).SwitchBoxY
1847                 'MsgBox "(" & NetTree(node).SwitchBoxX & "," &
1848                 NetTree(node).SwitchBoxY & ")"
1849                 Picture1.Line (8 + 32 * X1, 8 + 32 * Y1)-(8 + 32 * X2, 8
1850                 + 32 * Y2), RGB(0, 0, 255)
1851                 AddToChannels X1, Y1, X2, Y2, t
1852
1853                 X1 = X2
1854                 Y1 = Y2
1855             Loop
1856
1857             If node = 1 Then
1858                 LineStartEnd X1, Y1, p2, p2chha, True
1859                 If p2chha Then
1860                     AddToChannels X1, Y1, X1 + 1, Y1, t
1861                 Else
1862                     AddToChannels X1, Y1, X1, Y1 + 1, t
1863                 End If
1864             ElseIf node = 2 Then
1865                 LineStartEnd X1, Y1, p2, p2chha, False
1866                 If p2chha Then
1867                     AddToChannels X1 - 1, Y1, X1, Y1, t
1868                 Else
1869                     AddToChannels X1, Y1 - 1, X1, Y1, t
1870                 End If
1871             ElseIf node = 3 Then
1872                 LineStartEnd X1, Y1, p2, p2chhb, True
1873                 If p2chhb Then
1874                     AddToChannels X1, Y1, X1 + 1, Y1, t
1875                 Else
1876                     AddToChannels X1, Y1, X1, Y1 + 1, t
1877                 End If
1878             ElseIf node = 4 Then
1879                 LineStartEnd X1, Y1, p2, p2chhb, False
1880                 If p2chhb Then
1881                     AddToChannels X1 - 1, Y1, X1, Y1, t
1882                 Else
1883                     AddToChannels X1, Y1 - 1, X1, Y1, t
1884                 End If
1885             End If
1886
```

```
1887         'same thing with end point
1888
1889         If Not ToNet Then
1890
1891         With NetTree(CostMinNode)
1892
1893         X1 = .SwitchBoxX
1894         Y1 = .SwitchBoxY
1895
1896         If .FinalCostB = 0 Or (.FinalCostA > 0 And .FinalCostA <=
1897         .FinalCostB) Then
1898             'use A
1899             LineStartEnd X1, Y1, p1, plchha, CostMinTopLeft
1900         Else
1901             'use B
1902             LineStartEnd X1, Y1, p1, plchhb, CostMinTopLeft
1903         End If
1904
1905         End With
1906
1907         End If
1908
1909     End If
1910
1911     Picture1.Refresh
1912
1913     oldtimer = Timer
1914     Do Until Timer > oldtimer + 0.5
1915         DoEvents
1916     Loop
1917
1918 Next
1919 End Function
1920
1921 Private Sub AddToChannels(X1 As Integer, Y1 As Integer, X2 As Integer,
1922 Y2 As Integer, Net As Integer)
1923     Dim inuse As Boolean
1924
1925     inuse = False
1926     If X1 = X2 Then
1927         'V - use (x1,(y1+y2-1)/2)
1928         For n = 1 To VChannelNum(X1, (Y1 + Y2 - 1) / 2)
1929             If VChannel(X1, (Y1 + Y2 - 1) / 2, n) = Net Then
1930                 inuse = True
1931             End If
1932         Next
1933     If Not inuse Then
1934         'add net to the channel
1935         VChannelNum(X1, (Y1 + Y2 - 1) / 2) = VChannelNum(X1, (Y1 +
1936 Y2 - 1) / 2) + 1
1937         VChannel(X1, (Y1 + Y2 - 1) / 2, VChannelNum(X1, (Y1 + Y2 -
1938 1) / 2)) = Net
1939     End If
1940 Else
1941     'H - use ((x1+x2-1)/2,y1)
1942     For n = 1 To HChannelNum((X1 + X2 - 1) / 2, Y1)
1943         If HChannel((X1 + X2 - 1) / 2, Y1, n) = Net Then
1944             inuse = True
1945         End If
1946     Next
1947 If Not inuse Then
```

```
1945         'add net to channel
1946         HChannelNum((X1 + X2 - 1) / 2, Y1) = HChannelNum((X1 + X2 -
1947         1) / 2, Y1) + 1
1948         HChannel((X1 + X2 - 1) / 2, Y1, HChannelNum((X1 + X2 - 1) /
1949         2, Y1)) = Net
1950     End If
1951 End Sub
1952 Private Sub LineStartEnd(X As Integer, Y As Integer, Point As String, H
As Boolean, TopLeft As Boolean)
1953     'now make final node connections
1954     'plchha = true:
1955     '    node = 1: left of node, position A
1956     '    node = 2: right of node, position A
1957     'plchha = false:
1958     '    node = 1: top of node, position A
1959     '    node = 2: bottom of node, position A
1960     'plchhb = true:
1961     '    node = 3: left of node, position B
1962     '    node = 4: right of node, position B
1963     'plchhb = false:
1964     '    node = 3: top of node, position B
1965     '    node = 4: bottom of node, position B
1966     If TopLeft Then
1967         If H Then
1968             'left of node, position A
1969             Picture1.Line (8 + 32 * X, 8 + 32 * Y)-(40 + 32 * X, 8 + 32
1970             * Y), RGB(0, 0, 255)
1971             If Left(Point, 1) = "n" Or Left(Point, 1) = "s" Or
1972             (Left(Point, 1) = "P" And Val(Right(Point, 3)) >= 18) Then
1973                 'line down
1974                 Picture1.Line (24 + 32 * X, 8 + 32 * Y)-(24 + 32 * X, 24
1975                 + 32 * Y), RGB(0, 0, 255)
1976                 ElseIf Left(Point, 1) = "N" Or Left(Point, 1) = "S" Or
1977                 (Left(Point, 1) = "P" And Val(Right(Point, 3)) <= 5) Then
1978                     'line up
1979                     Picture1.Line (24 + 32 * X, 8 + 32 * Y)-(24 + 32 * X, 32
1980                     * Y - 16), RGB(0, 0, 255)
1981                 End If
1982             Else
1983                 'top of node, position A
1984                 Picture1.Line (8 + 32 * X, 8 + 32 * Y)-(8 + 32 * X, 40 + 32
1985                 * Y), RGB(0, 0, 255)
1986                 If Left(Point, 1) = "n" Or Left(Point, 1) = "s" Or
1987                 (Left(Point, 1) = "P" And Val(Right(Point, 3)) Mod 2 = 1) Then
1988                     'line right
1989                     Picture1.Line (8 + 32 * X, 24 + 32 * Y)-(24 + 32 * X, 24
1990                     + 32 * Y), RGB(0, 0, 255)
1991                     ElseIf Left(Point, 1) = "N" Or Left(Point, 1) = "S" Or
1992                     (Left(Point, 1) = "P" And Val(Right(Point, 3)) Mod 2 = 0) Then
1993                         'line left
1994                         Picture1.Line (8 + 32 * X, 24 + 32 * Y)-(32 * X - 8, 24
1995                         + 32 * Y), RGB(0, 0, 255)
1996                     End If
1997                 End If
1998             Else
1999                 If H Then
2000                     'right of node, position A
2001                     Picture1.Line (8 + 32 * X, 8 + 32 * Y)-(32 * X - 24, 8 + 32
2002                     * Y), RGB(0, 0, 255)
2003                     If Left(Point, 1) = "n" Or Left(Point, 1) = "s" Or
```

```

    (Left(Point, 1) = "P" And Val(Right(Point, 3)) >= 18) Then
1993     'line down
1994     Picture1.Line (32 * X - 8, 8 + 32 * Y)-(32 * X - 8, 24 +
32 * Y), RGB(0, 0, 255)
1995     ElseIf Left(Point, 1) = "N" Or Left(Point, 1) = "S" Or
(Left(Point, 1) = "P" And Val(Right(Point, 3)) <= 5) Then
1996     'line up
1997     Picture1.Line (32 * X - 8, 8 + 32 * Y)-(32 * X - 8, 32 *
Y - 16), RGB(0, 0, 255)
1998     End If
1999     Else
2000     'bottom of node, position A
2001     Picture1.Line (8 + 32 * X, 8 + 32 * Y)-(8 + 32 * X, 32 * Y -
24), RGB(0, 0, 255)
2002     If Left(Point, 1) = "n" Or Left(Point, 1) = "s" Or
(Left(Point, 1) = "P" And Val(Right(Point, 3)) Mod 2 = 1) Then
2003     'line right
2004     Picture1.Line (8 + 32 * X, 32 * Y - 8)-(24 + 32 * X, 32
* Y - 8), RGB(0, 0, 255)
2005     ElseIf Left(Point, 1) = "N" Or Left(Point, 1) = "S" Or
(Left(Point, 1) = "P" And Val(Right(Point, 3)) Mod 2 = 0) Then
2006     'line left
2007     Picture1.Line (8 + 32 * X, 32 * Y - 8)-(32 * X - 8, 32 *
Y - 8), RGB(0, 0, 255)
2008     End If
2009     End If
2010     End If
2011 End Sub
2012
2013 Private Sub AddNetTreeNode(Parent As Long, SwitchBoxX As Integer,
SwitchBoxY As Integer, ExtendedFrom As Integer, Cost As Integer)
2014     NetTreeLen = NetTreeLen + 1
2015
2016     'add to parent
2017     If Parent > 0 Then 'parent 0 is a root node
2018         With NetTree(Parent)
2019             .Children = .Children + 1
2020             .Child(.Children) = NetTreeLen
2021         End With
2022     End If
2023
2024     With NetTree(NetTreeLen)
2025         .Parent = Parent
2026         .Children = 0
2027         .ExtendedFrom = ExtendedFrom
2028         .SwitchBoxX = SwitchBoxX
2029         .SwitchBoxY = SwitchBoxY
2030         .Cost = Cost
2031     End With
2032
2033 End Sub
2034
2035 Private Function CheckVChannelTempB(X As Integer, Y As Integer)
2036     If 1 <= X And X <= 7 And 1 <= Y And Y <= 6 Then
2037         CheckVChannelTempB = VChannelTempB(X, Y)
2038     Else
2039         CheckVChannelTempB = False
2040     End If
2041 End Function
2042
2043 Private Function CheckHChannelTempB(X As Integer, Y As Integer)
2044     If 1 <= X And X <= 6 And 1 <= Y And Y <= 7 Then
```

```
2045         CheckHChannelTempB = HChannelTempB(X, Y)
2046     Else
2047         CheckHChannelTempB = False
2048     End If
2049 End Function
2050
2051 Private Sub Form_Load()
2052     StartEnd = True 'start
2053     n = 0
2054 End Sub
2055
2056
2057
2058
```