

```

Form1 - 1
Dim Gen_id%, dev_Osc, Command, param, Out_File, Out_Data, Volt_List, Temp_List, ReadBuffer As String
Dim SSComment As String
Dim Frequency(300), Voltage(300), Temperature(5000) As Single
Dim Freq, Amplitude, Offset, AmpGain, AvPer, VScal, VScalMax, Vmax As Single
Dim Temp_Wait, LastTemp, WaitV, WaitingVoltage, Accuracy, CurrentT, SetT As Single
Dim Num_Volt As Integer, Num_Temp As Integer, TemRes As Integer
Dim Fast As Boolean, AST As Boolean, ASV As Boolean, Expire As Boolean, DCmode As Boolean
Dim EndOfProgram As Label, da As Date

```

```

Private Sub Command1_Click()
'Load Form2 'Form2.Show
Command1.Enabled = False
Command2.Enabled = True
Dim Port As Integer
Dim Vmnt As Variant
Text15.Text = "Initiation"
DoEvents
'AvaSpec.ASactive ' establishing connection to the AVANTES software
Const AH As String = "AvaSoft© 7.5.3 Full - 2010 Avantes - S/N: 0502009S1"
AppActivate AH

Vmnt = MsgBox("Adjust the sensitivity and save the background", vbOKOnly + vbMsgBoxSetForeground)

```

```

Fill_Volt (Volt_List) 'Fill the measuring Voltage array
Fill_Temp (Temp_List) 'Fill the measuring Temperature array
Port = 1 'sign = 10
Init_COM (Port)
Init_Gen

```

```

Freq = Val(Text13.Text)
wrt_buf$ = "APPL:SQU " + Str(Freq)
ilwrt Gen_id%, wrt_buf$, Len(wrt_buf$)
Set_Amplitude (Vmax)
DCmode = False

```

```

Dim Rd, BaseName, Fold, FoldName, T_Name As String
Fold = Text7.Text
BaseName = Text8.Text
FoldName = Fold + BaseName

```

```

'***** BEGIN Temperature CICLE *****
it = 1
TemRes = 100
Do While Temperature(it) <> 999
SetT = Temperature(it)
T_Name = FoldName + "T" + Trim$(Str$(Int((Temperature(it) * TemRes) + 1 / TemRes)))
Out_Data = T_Name + ".dat"
Beep
Set_Temp (SetT)
If it <> 1 Then Waiting (Temp_Wait)
Text15.Text = "Waiting for Temperature"
DoEvents
CurrentT = Read_Temp
If Abs(SetT - CurrentT) > Accuracy And Text12.Text <> "" Then WaitTemp

```

```

'***** BEGIN Voltage CICLE *****
iv = 1
Do While Voltage(iv) <> 99999
Set_Amplitude (Voltage(iv) / AmpGain): Sleep (1000)
Text15.Text = "V circle": DoEvents
Sleep (WaitV)
'AvaSpec.SaveSpec ("T" + CStr(SetT) + "V" + CStr(Voltage(iv)))

```

```

SSComment = "T" + CStr(SetT) + "V" + CStr(Voltage(iv))
AppActivate AH
Sleep (300)
SendKeys "%(FSE)", True
Sleep (300)
SendKeys SSComment, True
Sleep (300)
SendKeys "{ENTER}", True
Sleep (300)

```

```

        iv = iv + 1
Loop   '***** END Of Voltage CICLE *****
If WaitingVoltage <> 0 Then Set_Amplitude (WaitingVoltage / AmpGain)

    it = it + 1
Loop   '***** END Of Temperature CICLE *****
If LastTemp <> 0 Then Set_Temp (LastTemp)
Close_COM
EndOfProgram:
End Sub   'END OF MAIN PROGRAM
Private Sub Waiting(min)
'Dim min As Single
Dim sec As Integer
Dim i As Integer
sec = Round(min * 60)
For i = 1 To sec
Sleep (1000)
Text15.Text = Mess$ + " ( time left: " + Str(Round(sec - i)) + " sec )"
DoEvents
Next i
End Sub
Private Sub WaitTemp()
Mess$ = "Waiting for Accuracy": Beep
Dim sec As Integer
Dim i As Integer
i = 0

Do While Abs(SetT - CurrentT) > Accuracy
Sleep (1000)
Text15.Text = Mess$ + Str(i) + " sec "
DoEvents
i = i + 1
CurrentT = Read_Temp
Loop
End Sub
Private Sub Fill_Volt(tlist) '*****
TL = Trim$(tlist)
p1 = 1
Vmax = 0
pos = 0: i1 = 0
Do While p1 > 0
    i1 = i1 + 1
    p1 = InStr(TL, ",") 'Get position of next comma.
    p2 = InStr(TL, "/") 'Get position of next "/".
    If p1 > 0 Or p2 > 0 Then 'Check for comma
        If p1 <> 0 Then
            v11 = Left$(TL, p1 - 1)
            TL = Mid$(TL, p1 + 1)
        Else
            v11 = TL
        End If
        p2 = InStr(v11, "/") 'Get position of next "/".
        If p2 <> 0 Then ' Check for "/"
            vol1 = Val(Left$(v11, p2 - 1))
            v11 = Mid$(v11, p2 + 1)
            p2 = InStr(v11, "/")
            vols = Val(Left$(v11, p2 - 1))
            vol2 = Val(Mid$(v11, p2 + 1))
            If vol1 > vol2 Then vols = -vols
            For vol = vol1 To vol2 Step vols
                Voltage(i1) = vol
                i1 = i1 + 1
            Next vol
            i1 = i1 - 1
        Else
            Voltage(i1) = Val(v11)
            If Voltage(i1) > Vmax Then Vmax = Voltage(i1)
        End If
    Else

```

```

        Voltage(i1) = Val(TL)
        If Voltage(i1) > Vmax Then Vmax = Voltage(i1)
    End If

    Loop
    Num_Volt = i1
    Voltage(i1 + 1) = 99999
End Sub

Private Sub Fill_Temp(tlist) '*****
TL = Trim$(tlist)
p5 = 1

pos = 0: i5 = 0
Do While p5 > 0
    i5 = i5 + 1
    p5 = InStr(TL, ",") 'Get position of next comma.
    p6 = InStr(TL, "/") 'Get position of next "/".
    If p5 > 0 Or p6 > 0 Then 'Check for comma
        If p5 <> 0 Then
            TL1 = Left$(TL, p5 - 1)
            TL = Mid$(TL, p5 + 1)
        Else
            TL1 = TL
        End If
        p6 = InStr(TL1, "/") 'Get position of next "/".
        If p6 <> 0 Then ' Check for "/"
            tem1 = Val(Left$(TL1, p6 - 1))
            TL1 = Mid$(TL1, p6 + 1)
            p6 = InStr(TL1, "/")
            tems = Val(Left$(TL1, p6 - 1))
            tem2 = Val(Mid$(TL1, p6 + 1))
            If tem1 > tem2 Then tems = -tems
            For vol = tem1 To tem2 Step tems
                Temperature(i5) = vol
                i5 = i5 + 1
            Next vol
            i5 = i5 - 1
        Else
            Temperature(i5) = Val(TL1)
        End If
    Else
        Temperature(i5) = Val(TL)
    End If

    Loop
    Num_Temp = i5
    Temperature(i5 + 1) = 999
End Sub

'*****Commands for Oscilloscope*****

'*****Commands for Generator*****

Private Sub Set_WaveForm(Form)
wrt_buf$ = "FUNC:SHAP " + Form
ilwrt Gen_id%, wrt_buf$, Len(wrt_buf$)
End Sub

Private Sub Set_Freq(Freq)
wrt_buf$ = "FREQ " + Str(Freq)
ilwrt Gen_id%, wrt_buf$, Len(wrt_buf$)
End Sub

Private Sub Set_Amplitude(Amplitude)
If Amplitude <> 0 Then
    If DCmode Then
        wrt_buf$ = "APPL:SQU " + Str(Freq)
        ilwrt Gen_id%, wrt_buf$, Len(wrt_buf$)
        wrt_buf$ = "VOLT " + Str(Amplitude)
        ilwrt Gen_id%, wrt_buf$, Len(wrt_buf$)
        DCmode = False
    Else
        wrt_buf$ = "VOLT " + Str(Amplitude)
        ilwrt Gen_id%, wrt_buf$, Len(wrt_buf$)
    End If
End If

```

```

        End If
        Else
            DCmode = True
            wrt_buf$ = "APPLY:DC DEF, DEF, " + Str(Offset)
            ilwrt Gen_id%, wrt_buf$, Len(wrt_buf$)
End If

End Sub

Private Sub Set_Offset(Offset)
    wrt_buf$ = "VOLT:OFFS " + Str(Offset)
    ilwrt Gen_id%, wrt_buf$, Len(wrt_buf$)
End Sub

Private Sub Init_Gen()
    Const B_G = 0
    Const P_G = 10
    Const N_G = 0
    Const T_G = Tls
    Const E1_G = 1
    Const E2_G = 0
    Gen_id% = ildev(B_G, P_G, N_G, T_G, E1_G, E2_G)
    Set_Offset ("0")
End Sub

Private Sub Init_COM(Port) '*****
'da = "01/09/2017" ' If Expire = True Then GoTo EndOfProgram
'If Date > da Then Expire = True
MSComm1.CommPort = Port
MSComm1.Settings = "9600,N,8,1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True
End Sub

Private Sub Close_COM()
MSComm1.PortOpen = False
End Sub

Private Function Set_Temp(temp)
    temp = TemRes * temp
    Dim ADDRESS, CODE, A_MSB, A_LSB, V_MSB, V_LSB As Byte
    ADDRESS = 1
    CODE = 6
    A_MSB = 0
    A_LSB = 2
    V_MSB = temp \ 256
    V_LSB = temp Mod 256
    message$ = Chr(ADDRESS) + Chr(CODE) + Chr(A_MSB) + Chr(A_LSB) + Chr(V_MSB) + Chr(V_LSB)
    err1 = CRC(message$)
    MSComm1.Output = message$
    Set_Temp = 1#
End Function

Private Function Read_Temp() As Double
    ADDRESS = 1
    CODE = 3
    A1_H = 0
    A1_L = 1 ' 1- Display; 2-SetPoint
    N_H = 0
    N_L = 1
    Mes$ = MSComm1.Input ' Clear the buffer
    Sleep (100)
    message$ = Chr(ADDRESS) + Chr(CODE) + Chr(A1_H) + Chr(A1_L) + Chr(N_H) + Chr(N_L)
    err1 = CRC(message$)
    MSComm1.Output = message$
    Sleep (100)
    Mes$ = MSComm1.Input
    Read_Temp = (256 * Asc(Mid(Mes$, 4, 1)) + Asc(Mid(Mes$, 5, 1))) / TemRes
End Function

Function CRC(message$) As Long
    CRC16& = 65535
    For C% = 1 To Len(message$)
        CRC16& = CRC16& Xor Asc(Mid$(message$, C%, 1))
        For Bit% = 1 To 8
            If CRC16& Mod 2 Then
                CRC16& = (CRC16& \ 2) Xor 40961
            Else
                CRC16& = CRC16& \ 2
            End If
        Next Bit%
    Next C%
End Function

```

```

        End If
    Next Bit%
Next C%
CRCH% = CRC16% \ 256: CRCL% = CRC16% Mod 256
message$ = message$ + Chr$(CRCL%) + Chr$(CRCH%) + "xyz"
CRC = CRC16%
End Function

Private Sub Command2_Click()
'Unload Form2
Unload Form1
End Sub

Private Sub Form_Load() '*****
TemRes = Val(Text1.Text)
' = Val(Text2.Text)
AvPer = Val(Text3.Text)
Volt_List = Text4.Text
Temp_List = Text6.Text
Accuracy = Round(Val(Text12.Text), 2)
Offset = Val(Text5.Text)
Temp_Wait = Val(Text9.Text)
' = Val(Text10.Text)
VScalMax = Val(Text14.Text)
AmpGain = Round(Val(Text17.Text))
WaitV = Val(Text18.Text) * 1000
Expire = False
'Form2.Picture1.Cls
'Form2.Picture1.BackColor = RGB(255, 255, 255)

AST = Check2.value
ASV = Check4.value

End Sub

Private Sub Form_Unload(Cancel As Integer)
End
End Sub

Private Sub Text1_Change()
TemRes = Val(Text1.Text)
End Sub

Private Sub Text10_Change()
'QQ = Int(Val(Text10.Text))

End Sub

Private Sub Text11_Change()
LastTemp = Val(Text11.Text)
End Sub

Private Sub Text12_Change()
Accuracy = Round(Val(Text12.Text), 2)
End Sub

Private Sub Text14_Change()
VScalMax = Val(Text14.Text)
End Sub

Private Sub Text16_Change()
WaitingVoltage = Round(Val(Text16.Text), 1)
End Sub

Private Sub Text17_Change()
AmpGain = Round(Val(Text17.Text))
End Sub

Private Sub Text18_Change()
WaitV = Val(Text18.Text) * 1000
End Sub

Private Sub Text2_Change()
' = Val(Text2.Text)
End Sub

Private Sub Text3_Change()
AvPer = Val(Text3.Text)
End Sub

Private Sub Text4_Change()
Volt_List = Text4.Text

```

```
End Sub
Private Sub Text5_Change()
Offset = Val(Text6.Text)
End Sub
Private Sub Text6_Change()
Temp_List = Text6.Text
End Sub
Private Sub Text7_Change()
Fold = Text7.Text
End Sub
Private Sub Text9_Change()
Temp_Wait = Val(Text9.Text)
End Sub
Private Sub Check1_Click()
' = Check1.value
End Sub
Private Sub Check2_Click()
AST = Check2.value
End Sub
Private Sub Check4_Click()
ASV = Check4.value
End Sub
```