#pragma rtGlobals=1

// \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//

// Amperometric Spike Analysis

// Version 8.15 06/12/2005

//

// Eugene Mosharov, PhD

// Columbia University, Dpt. of Neurology

// em706@columbia.edu or johnmosh@yahoo.com

// \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// To start:

// 1. Hide or Minimize this window

// 2. Choose \* Prepare for Quanta Analysis \* from Macros menu

Menu "Macros"

"-"

"\* Prepare for Quanta Analysis! \*",Prepare\_for\_Trace\_Analysis()

End

Macro Prepare\_for\_Trace\_Analysis()

String Existing\_windows=WinList("\*", ";","WIN:83") // A list of all windows, panels and notebooks

String One\_win=StringFromList(0, Existing\_windows)

If(strlen(One\_win))

DoAlert 1,"Delete existing Windows?"

If(V\_Flag==1)

Variable i=0

Do

One\_win=StringFromList(i, Existing\_windows)

If(strlen(One\_win)==0)

break

endif

DoWindow/K $One\_win

i+=1

while(1)

endif

endif

if (DataFolderExists("root:Quanta"))

SetDataFolder $"root:Quanta"

else

NewDataFolder/S root:Quanta

endif

// Default values of global variables

if (CheckName("Bkg\_noise\_I", 3)==0)

Variable/G Bkg\_noise\_I=0 // Standard Deviation of the noise on non-differentiated trace. Used during foot detection.

endif

if (CheckName("Bkg\_noise\_dI", 3)==0)

Variable/G Bkg\_noise\_dI=0 // Standard Deviation of the noise on differentiated trace. Used for spike detection

endif

if (CheckName("Bkg\_noise\_Start", 3)==0)

Variable/G Bkg\_noise\_Start=0 // Beginning of the segment used to calculate SD of the noise.

endif

if (CheckName("Bkg\_noise\_End", 3)==0)

Variable/G Bkg\_noise\_End=0 // End of the segment used to calculate SD of the noise.

endif

if (CheckName("Detection\_Mult", 3)==0)

Variable/G Detection\_Mult=5 // Theshold for spike detection

endif

if (CheckName("Detection\_Foot\_Mult", 3)==0)

Variable/G Detection\_Foot\_Mult=2 // Theshold for foot detection

endif

if (CheckName("Smoothing\_Factor", 3)==0)

Variable/G Smoothing\_Factor=600 // Cutoff of the filter used on non-differentiated trace. Can be changed from the main window.

endif

if (CheckName("Smoothing\_Factor\_diff1", 3)==0)

Variable/G Smoothing\_Factor\_diff1=300 // Cutoff of Gaussian filter used on differentiated trace. Can be changed from Filters/Scales panel.

endif

if (CheckName("Smoothing\_Factor\_Add", 3)==0)

Variable/G Smoothing\_Factor\_Add=150 // Cutoff of additional Gaussian filter used on non-differentiated trace. Can be changed from Filters/Scales panel.

endif

if (CheckName("Smooth\_Derivative", 3)==0)

Variable/G Smooth\_Derivative=1 // Allows filtering of differentiated trace.

endif

if (CheckName("Smooth\_more", 3)==0)

Variable/G Smooth\_more=0 // Allows additional filtering of non-differentiated trace.

endif

if (CheckName("Overall\_Filter", 3)==0)

Variable/G Overall\_Filter=0 // Overall additive filtering (except additional) applied to non-differentiated trace.

endif

if (CheckName("Spike\_Min\_Imax", 3)==0)

Variable/G Spike\_Min\_Imax=0 // I(max) cutoff

endif

if (CheckName("Spike\_Min\_Imax\_Last", 3)==0)

Variable/G Spike\_Min\_Imax\_Last=3 // Last used I(max) cutoff

endif

if (CheckName("Spike\_Max\_T05", 3)==0)

Variable/G Spike\_Max\_T05=0 // t(1/2) cutoff

endif

if (CheckName("Spike\_Max\_T05\_Last", 3)==0)

Variable/G Spike\_Max\_T05\_Last=1 // Last used t(1/2) cutoff

endif

if (CheckName("Spike\_Max\_Trise", 3)==0)

Variable/G Spike\_Max\_Trise=0 // t(rise) cutoff

endif

if (CheckName("Spike\_Max\_Trise\_Last", 3)==0)

Variable/G Spike\_Max\_Trise\_Last=2 // Last used t(rise) cutoff

endif

if (CheckName("Foot\_Min\_W", 3)==0)

Variable/G Foot\_Min\_W=0 // T(foot) cutoff

endif

if (CheckName("Foot\_Min\_W\_Last", 3)==0)

Variable/G Foot\_Min\_W\_Last=2 // Last used T(foot) cutoff

endif

if (CheckName("SSFoot\_Do", 3)==0)

Variable/G SSFoot\_Do=0 // Delete feet without steady states

endif

if (CheckName("Native\_Foot\_Del", 3)==0)

Variable/G Native\_Foot\_Del=0 // Delete feet longer that 0.33\*t(rise 50-90%)

endif

if (CheckName("Foot\_Min\_H", 3)==0)

Variable/G Foot\_Min\_H=0 // I(foot) cutoff

endif

if (CheckName("Foot\_Min\_H\_Last", 3)==0)

Variable/G Foot\_Min\_H\_Last=1 // Last used I(foot) cutoff

endif

if (CheckName("Peak\_ID", 3)==0)

Variable/G Peak\_ID=0 // The ID of the currently active spike

endif

if (CheckName("Total\_peaks\_number", 3)==0)

Variable/G Total\_peaks\_number=0 // Total number of detected spikes

endif

if (CheckName("Rise\_Low\_Prc", 3)==0)

Variable/G Rise\_Low\_Prc=25 // The lower point (% of Imax) on spike rising phase used to calculate r(rise).

endif

if (CheckName("Rise\_Hi\_Prc", 3)==0)

Variable/G Rise\_Hi\_Prc=75 // The upper point (% of Imax) on spike rising phase used to calculate r(rise).

endif

if (CheckName("Fit\_method", 4)==0)

String/G Fit\_method="DblExp" // Function used to fit the falling phase of the spikes

endif

if (CheckName("Fall\_ChiRatio\_Cutoff", 3)==0)

variable/G Fall\_ChiRatio\_Cutoff=2 // Chi^2(double exp) to Chi^2(single exp) ratio threshold for the DblExp fit.

endif

if (CheckName("Gain", 3)==0)

variable/G Gain=1 // Trace Y scaling gain.

endif

if (CheckName("Gain\_Temp", 3)==0)

variable/G Gain\_Temp=1 // Trace Y scaling gain. Additional variable used to determine if the gain has been changed.

endif

if (CheckName("Zoom\_On", 4)==0)

String/G Zoom\_On="Off" // Zooms in and out of the currently active spike in the main window.

endif

if (CheckName("Show\_Legend", 3)==0)

variable/G Show\_Legend=1 // Enables the legend on top of the main window.

endif

if (CheckName("T\_Start\_orig", 3)==0)

Variable/G T\_Start\_orig=0 // X value of the first datapoint in the original trace.

endif

if (CheckName("T\_Delta\_orig", 3)==0)

variable/G T\_Delta\_orig=0.1 // Sampling interval (microS) of the original trace.

endif

if (CheckName("T\_Start", 3)==0)

Variable/G T\_Start=0 // X value of the first datapoint in the working trace copy.

endif

if (CheckName("T\_End", 3)==0)

Variable/G T\_End=0 // X value of the last datapoint in the working trace copy.

endif

if (CheckName("T\_Delta", 3)==0)

variable/G T\_Delta=0.1 // Sampling interval (microS) of the working trace copy.

endif

if (CheckName("X\_min", 3)==0)

Variable/G X\_min=0 // X value of the first datapoint displayed in the main window.

endif

if (CheckName("X\_max", 3)==0)

Variable/G X\_max=0 // X value of the last datapoint displayed in the main window.

endif

if (CheckName("Y\_min", 3)==0)

Variable/G Y\_min=0 // Minimal Y value displayed in the main window.

endif

if (CheckName("Y\_max", 3)==0)

Variable/G Y\_max=0 // Maximal Y value displayed in the main window.

endif

if (CheckName("Population\_Center", 4)==0)

String/G Population\_Center="Median" // Calculate a Mean or a Median of spike parameters during statistical analysis

endif

if (CheckName("Norm\_point", 4)==0)

String/G Norm\_point="Rise" // Uses spike Tmax ('Max') or the midpoint of the linear segment of the rising phase ('Rise') during spike averaging

endif

if (CheckName("Baseline\_Drift", 3)==0)

Variable/G Baseline\_Drift=50 // Maximal allowed baseline drift. If higher, the spike is deleted.

endif

if (CheckName("Overlap\_Prc", 3)==0)

Variable/G Overlap\_Prc=50 // Maximal allowed degree of spike overlap. If higher, both spikes are deleted.

endif

if (CheckName("Overlaps", 4)==0)

String/G Overlaps="Separate" // The remaining overlaps are either 'Ignored', 'Separated' or 'Split'.

endif

if (CheckName("File\_list", 4)==0)

String/G File\_list="" // The list of recently opened recordings

endif

if (CheckName("Loaded\_file\_path", 4)==0)

String/G Loaded\_file\_path="" // The name of the last opened recording

endif

if (CheckName("Fall\_Tau\_Extrap", 3)==0)

Variable/G Fall\_Tau\_Extrap=0 // Exponential Tau used for falling phase extrapolasion of overlapping spikes.

endif

// The list of parameters that will be displayed (not calculated) by default. To set new default settings change "0" to "1".

Variable/G Show\_Time=1 // T\_Max

Variable/G Show\_Base=0 // Peak\_Base

Variable/G Show\_Width=1 // Peak\_t05

Variable/G Show\_H=1 // Peak\_Imax

Variable/G Show\_Q=0 // Peak\_Q

Variable/G Show\_Molec=1 // Peak\_Molec

Variable/G Show\_Rise\_t=1 // Rise\_time

Variable/G Show\_Rise\_r=1 // Rise\_slope

Variable/G Show\_Fall\_t=1 // Fall\_time

Variable/G Show\_Fall\_r=1 // Fall\_slope

Variable/G Show\_Ft\_H=1 // Foot\_I

Variable/G Show\_Ft\_width=1 // Foot\_W

Variable/G Show\_Ft\_Q=0 // Foot\_Q

Variable/G Show\_Ft\_molec=1 // Foot\_Molec

String/G Values\_to\_show

String/G Stats\_names

// Build new windows

if (CheckName("Working\_trace\_copy", 1)==0)

Make/O/N=1 Working\_trace\_copy,Orig\_trace\_copy,diff1

Make/O/N=1 Zoomed\_peak, Fall\_phase,Rise\_phase

Make/O/N=0 Avg\_peak

endif

if (CheckName("Peak\_Num", 1)==0)

Change\_waves("Make",0)

endif

if (CheckName("File\_Q", 6)==0)

File\_Q()

else

Dowindow/F File\_Q

endif

Variable Resolution=ScreenResolution

String scrn=StringByKey("SCREEN1", IgorInfo(0))

Variable Scrn\_width=str2num(StringFromList(3, scrn,","))

Variable Scrn\_hight=str2num(StringFromList(4, scrn,","))

Variable/G Scrn\_width\_Points=Scrn\_width\*0.74609375\*96/Resolution

Variable/G Scrn\_hight\_Points= Scrn\_hight\*0.74609375\*96/Resolution

Variable Zoom\_Win\_width=220\*96/Resolution

Variable Table\_Win\_hight=180\*96/Resolution

if (CheckName("Main\_window", 6)==0)

execute "Main\_window()"

SetVariable ID,limits={1,(Total\_peaks\_number),1}

PauseUpdate; Silent 1

Movewindow 0,80,Scrn\_width\_Points-Zoom\_Win\_width,Scrn\_hight\_Points-Table\_Win\_hight

AutoPositionWindow/E/M=1/R=File\_Q Main\_window

Dowindow/F File\_Q

Show\_Extras("Zoom\_off")

MoveWindow /C Scrn\_width\_Points-Zoom\_Win\_width,320\*96/Resolution,Scrn\_width\_Points,Scrn\_hight\_Points-75\*96/Resolution

endif

if (CheckName("Peak\_stats\_Table1", 6)==0)

Change\_Table ("ctrlName",1)

Dowindow/F Peak\_stats\_Table1

AutoPositionWindow/E/M=0/R=Main\_window Peak\_stats\_Table1

endif

Endmacro

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_\_Spike Detection and Analysis\_\_\_\_\_\_

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Function Bkg\_noise\_Set(ctrlName) : ButtonControl

String ctrlname

SetDataFolder $"root:Quanta"

NVAR Bkg\_noise\_Start=Bkg\_noise\_Start

NVAR Bkg\_noise\_End=Bkg\_noise\_End

Wave Working\_trace\_copy=Working\_trace\_copy

If(numpnts(Working\_trace\_copy)<2)

abort "No trace loaded or the wave is too short."

endif

Bkg\_noise\_Start=min(xcsr(A), xcsr(B))

Bkg\_noise\_End=max(xcsr(A), xcsr(B))

Bkg\_noise\_Calc()

End

Function Bkg\_noise\_Calc()

SetDataFolder $"root:Quanta"

Wave Working\_trace\_copy=Working\_trace\_copy

NVAR Bkg\_noise\_I=Bkg\_noise\_I

NVAR Bkg\_noise\_dI=Bkg\_noise\_dI

NVAR Bkg\_noise\_Start=Bkg\_noise\_Start

NVAR Bkg\_noise\_End=Bkg\_noise\_End

NVAR Smoothing\_Factor\_diff1=Smoothing\_Factor\_diff1

NVAR Smooth\_more=Smooth\_more

NVAR Smooth\_Derivative=Smooth\_Derivative

NVAR Smoothing\_Factor\_Add=Smoothing\_Factor\_Add

If (Bkg\_noise\_Start==Bkg\_noise\_End)

String Note="Background level of noise is not set!\r"

Note+="Use cursors A and B to select an area of your trace\r"

Note+="that does not have any spikes and press 'Bkg' button"

abort Note

endif

wavestats/Q/R = (Bkg\_noise\_Start, Bkg\_noise\_End ) Working\_trace\_copy

Bkg\_noise\_I=V\_sdev

Duplicate/O/R=(Bkg\_noise\_Start,Bkg\_noise\_End) Working\_trace\_copy, Bkg\_diff1

Variable Binomial\_coeff

If (Smooth\_more==1)

Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Smoothing\_Factor\_Add)

Smooth Binomial\_coeff, Bkg\_diff1

endif

Differentiate Bkg\_diff1

If (Smooth\_Derivative==1)

Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Smoothing\_Factor\_diff1)

Smooth Binomial\_coeff, Bkg\_diff1

endif

wavestats/Q Bkg\_diff1

Bkg\_noise\_dI=V\_sdev

Killwaves/Z Bkg\_diff1

GroupBox Bkg\_HiLt, win=Main\_window,disable=1

End

Function Peak\_finder(ctrlName) : ButtonControl

String ctrlname

SetDataFolder $"root:Quanta"

Wave Working\_trace\_copy=Working\_trace\_copy

NVAR Detection\_Mult=Detection\_Mult

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Peak\_ID=Peak\_ID

NVAR Bkg\_noise\_dI=Bkg\_noise\_dI

NVAR Bkg\_noise\_Start=Bkg\_noise\_Start

NVAR Bkg\_noise\_End=Bkg\_noise\_End

NVAR Smoothing\_Factor=Smoothing\_Factor

NVAR Smoothing\_Factor\_diff1=Smoothing\_Factor\_diff1

NVAR Smooth\_Derivative=Smooth\_Derivative

NVAR Smooth\_more=Smooth\_more

NVAR Smoothing\_Factor\_Add=Smoothing\_Factor\_Add

NVAR Baseline\_Drift=Baseline\_Drift

Wave Peak\_Num=Peak\_Num

Wave T\_Max=T\_Max

Wave Peak\_Imax=Peak\_Imax

Wave Peak\_Half\_H2=Peak\_Half\_H2

Wave Peak\_Q=Peak\_Q

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Working\_trace\_copy=Working\_trace\_copy

If(numpnts(Working\_trace\_copy)<2)

abort "No trace loaded or the wave is too short."

endif

If(Total\_peaks\_number)

DoAlert 1, "All existing spikes will be deleted! \rPress 'Yes' to continue or 'No' to cancel."

If (V\_Flag==1)

De\_novo()

else

abort

endif

endif

Bkg\_noise\_Calc()

Getaxis/Q bottom

Variable Left\_X=V\_min

Variable Start\_X=V\_min

Variable End\_X=min(V\_max, pnt2x(Working\_trace\_copy,(numpnts(Working\_trace\_copy)-1)))

if (CheckName("Peak\_stats\_Table1", 6)==0)

execute "Peak\_stats\_Table()"

else

Dowindow/F Peak\_stats\_Table1

MoveWindow 1, 1, 1, 1

endif

Variable Binomial\_coeff,Real\_cutoff

If (Smooth\_more==1)

Duplicate/O/R=(Start\_X,End\_X) Working\_trace\_copy, diff1, diff1\_nosmooth

Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Smoothing\_Factor\_Add)

Smooth Binomial\_coeff, diff1

Real\_cutoff=Binomial\_to\_Gaussian\_Calc(Binomial\_coeff)

Print "Trace was additionally filtered with "+num2str(Real\_cutoff)+"Hz (Binomial "+num2str(Binomial\_coeff)+") -3dB Gaussian filter."

else

Duplicate/O/R=(Start\_X,End\_X) Working\_trace\_copy, diff1, diff1\_nosmooth

endif

Differentiate diff1, diff1\_nosmooth

If (Smooth\_Derivative==1)

Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Smoothing\_Factor\_diff1)

Smooth Binomial\_coeff, diff1, diff1\_nosmooth

endif

Variable Detection\_level\_diff1=Bkg\_noise\_dI\*Detection\_Mult

Total\_peaks\_number=0

Variable Kapec=0

Variable dI\_zero\_L, dI\_zero\_R,Limit\_R,Limit\_L

do

If (cmpnum(Start\_X,End\_X,18)!=0)

FindPeak/Q/M=(Detection\_level\_diff1)/R=(Start\_X,End\_X) diff1

else

Kapec=1

endif

if ((V\_Flag==0)%&(Kapec!=1))

If(V\_TrailingEdgeLoc>0)

Total\_peaks\_number+=1

Change\_waves("Redimension",Total\_peaks\_number)

Peak\_Num[Total\_peaks\_number-1]=Total\_peaks\_number

Start\_X=V\_PeakLoc

dI\_zero\_R=V\_TrailingEdgeLoc

dI\_zero\_L=V\_PeakLoc

Limit\_L=max(Left\_X,T\_Bkg2[Total\_peaks\_number-2])

// temporary Tmax

FindLevel/Q/R=(Start\_X,End\_X) diff1, 0

T\_Max[Total\_peaks\_number-1]=V\_LevelX

// next spike Tmax or the end of the trace

FindPeak/Q/M=(Detection\_level\_diff1)/R=(T\_Max[Total\_peaks\_number-1],End\_X) diff1

if (V\_Flag==0)

Limit\_R=V\_PeakLoc

else

Limit\_R=End\_X

endif

// final Tmax

FindLevel/Q/R=(T\_Max[Total\_peaks\_number-1],Limit\_R) Working\_trace\_copy, Working\_trace\_copy(dI\_zero\_L)

If(V\_flag==0)

dI\_zero\_R=V\_LevelX

wavestats/Q/R=(dI\_zero\_L,dI\_zero\_R) Working\_trace\_copy

T\_Max[Total\_peaks\_number-1]=V\_maxloc

endif

Cursor /W=Main\_window A Working\_trace\_copy T\_Max[Total\_peaks\_number-1]

// temporary Tbkg1

FindLevel/Q/R=(dI\_zero\_L,Limit\_L) diff1, 0

If(V\_flag==1)

T\_Bkg1[Total\_peaks\_number-1]=Limit\_L

else

T\_Bkg1[Total\_peaks\_number-1]=max(V\_LevelX,T\_Bkg2[Total\_peaks\_number-2])

endif

// final Tbkg1

Variable SteadyState=Find\_SteadyState(dI\_zero\_L, Limit\_L, 2\*(dI\_zero\_R-dI\_zero\_L))

If(SteadyState)

Variable SteadyState4=Find\_SteadyState(dI\_zero\_L, Limit\_L, 4\*(dI\_zero\_R-dI\_zero\_L))

If(SteadyState4)

FindLevel/Q/R=(dI\_zero\_L,Limit\_L) Working\_trace\_copy, SteadyState4

else

FindLevel/Q/R=(dI\_zero\_L,Limit\_L) Working\_trace\_copy, SteadyState

endif

T\_Bkg1[Total\_peaks\_number-1]=V\_LevelX

endif

Start\_X=dI\_zero\_R

// Tbkg2

FindLevel /Q/R=(T\_Max[Total\_peaks\_number-1],Limit\_R) diff1, 0

If (V\_Flag)

Peak\_Q[Total\_peaks\_number-1]=0

else

T\_Bkg2[Total\_peaks\_number-1]=V\_LevelX

Variable Flat\_Peak\_End=Find\_level\_plus\_2SD(T\_Max[Total\_peaks\_number-1],Limit\_R,T\_Bkg1[Total\_peaks\_number-1])

If(Flat\_Peak\_End)

T\_Bkg2[Total\_peaks\_number-1]=max(Flat\_Peak\_End,T\_Bkg2[Total\_peaks\_number-1])

else

wavestats/Q/R=(T\_Max[Total\_peaks\_number-1],Limit\_R) Working\_trace\_copy

T\_Bkg2[Total\_peaks\_number-1]=V\_minloc

endif

If ((T\_Max[Total\_peaks\_number-1]-T\_Max[Total\_peaks\_number-2]>0)%|(Total\_peaks\_number==1))

Calc\_Peak\_Parameters(Total\_peaks\_number,T\_Bkg1[Total\_peaks\_number-1],T\_Bkg2[Total\_peaks\_number-1])

else

Peak\_Q[Total\_peaks\_number-1]=0

endif

Wavestats/Q Peak\_Q

If(V\_numNans!=0)

Peak\_Q[Total\_peaks\_number-1]=0

endif

If ((T\_Bkg1[Total\_peaks\_number-1]>=T\_Max[Total\_peaks\_number-1])%|(T\_Bkg2[Total\_peaks\_number-1]<=T\_Max[Total\_peaks\_number-1]))

Peak\_Q[Total\_peaks\_number-1]=0

endif

// Check for spikes on uneven baseline

Variable Spike\_Min\_Imax=Working\_trace\_copy(T\_Max[Total\_peaks\_number-1])-max(Working\_trace\_copy(T\_Bkg1[Total\_peaks\_number-1]),Working\_trace\_copy(T\_Bkg2[Total\_peaks\_number-1]))

Variable Spike\_Max\_Imax=Working\_trace\_copy(T\_Max[Total\_peaks\_number-1])-min(Working\_trace\_copy(T\_Bkg1[Total\_peaks\_number-1]),Working\_trace\_copy(T\_Bkg2[Total\_peaks\_number-1]))

If(Spike\_Min\_Imax<(Spike\_Max\_Imax\*(100-Baseline\_Drift)/100))

Peak\_Q[Total\_peaks\_number-1]=0

endif

endif

Variable Next\_Start\_X=0

If(Detection\_limits(Total\_peaks\_number))

If(Total\_peaks\_number>1)

Next\_Start\_X=Peak\_Half\_H2[Total\_peaks\_number-1]

endif

Change\_waves("Delete",(Total\_peaks\_number-1))

Total\_peaks\_number=Total\_peaks\_number-1

endif

Start\_X=max(Start\_X,Next\_Start\_X)

else

Start\_X=V\_PeakLoc

endif

else

SVAR Overlaps

if((cmpstr(Overlaps,"Ignore")!=0)&(Total\_peaks\_number>1))

Check\_for\_Overlaps()

endif

if(Total\_peaks\_number==0)

Peak\_ID=0

SetVariable ID,limits={0,0,0},win=Main\_window

if (CheckName("Zoom\_Win", 6)!=0)

SetVariable ID,limits={0,0,0},win=Zoom\_Win

endif

SetDrawLayer /K UserFront

abort "No amperometric events found!"

else

Peak\_ID=1

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Main\_window

if (CheckName("Zoom\_Win", 6)!=0)

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Zoom\_Win

endif

Peak\_locator(0)

print "Found "+num2str(Total\_peaks\_number)+" spikes."

Abort

endif

endif

while (1)

End

Function Check\_for\_Overlaps()

SetDataFolder $"root:Quanta"

Wave Working\_trace\_copy=Working\_trace\_copy

Wave T\_Max=T\_Max

Wave Peak\_Num=Peak\_Num

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Peak\_t05=Peak\_t05

Wave/T Fall\_fit=Fall\_fit

NVAR Overlap\_Prc=Overlap\_Prc

SVAR Overlaps=Overlaps

NVAR Bkg\_noise\_I=Bkg\_noise\_I

NVAR Total\_peaks\_number=Total\_peaks\_number

// Two spikes are considered overlapping if the distance between the end of the 1st spike and the beginning of the 2nd one is less than the average t1/2 \* 2.

Variable i=1, ii=1

Variable Last\_overlapping\_peak

Variable Peak\_limit

Variable Same\_level

Make/O/N=(Total\_peaks\_number) Peak\_Delta

Peak\_Delta[1,]=((T\_Bkg1[p]-T\_Bkg2[p-1])-2\*(mean(Peak\_t05, 0, Total\_peaks\_number)/1000))\*1000

do

if((Working\_trace\_copy(T\_Bkg2[i-1])>(Working\_trace\_copy(T\_Bkg1[i-1])+2\*Bkg\_noise\_I))&(Peak\_Delta[i]<0))

FindLevel /P/Q/R=[i,Total\_peaks\_number] Peak\_Delta, 0

If(V\_LevelX)

Last\_overlapping\_peak=trunc(V\_LevelX)+1

else

Last\_overlapping\_peak=Total\_peaks\_number

endif

If((Last\_overlapping\_peak+1)<=Total\_peaks\_number)

Peak\_limit=T\_Bkg1[Last\_overlapping\_peak]

else

Getaxis/W=Main\_Window/Q bottom

Peak\_limit=min(V\_max, pnt2x(Working\_trace\_copy,(numpnts(Working\_trace\_copy)-1)))

endif

Same\_level=Find\_level\_plus\_2SD(T\_Max[Last\_overlapping\_peak-1],Peak\_limit,T\_Bkg1[i-1])

Variable Final\_level

if(Same\_level)

Final\_level=Same\_level

else

Final\_level=T\_Bkg2[Last\_overlapping\_peak-1]

endif

T\_Bkg2[i-1,Last\_overlapping\_peak-1]=Final\_level

ii=i-1

Variable Seddle, Min\_Tmax, Kill\_it=0

do

Wavestats/Q/R=(T\_max[ii],T\_max[ii+1]) Working\_trace\_copy

Seddle=V\_min-Working\_trace\_copy(Final\_level)

Min\_Tmax=min(Working\_trace\_copy(T\_max[ii])-Working\_trace\_copy(Final\_level),Working\_trace\_copy(T\_max[ii+1])-Working\_trace\_copy(Final\_level))

If((Seddle/Min\_Tmax\*100)>Overlap\_Prc)

Kill\_it=1

endif

ii+=1

while (ii<Last\_overlapping\_peak-1)

if((cmpstr(Overlaps,"Delete")==0)%|(Kill\_it))

print Last\_overlapping\_peak-(i-1),"overlapping spikes at",T\_Max[i-1],"were deleted"

ii=i-1

Do

Change\_waves("Delete",i-1)

DeletePoints i-1,1,Peak\_Delta

Peak\_Num[i-1,]-=1

ii+=1

Total\_peaks\_number-=1

while(ii<Last\_overlapping\_peak)

i-=Last\_overlapping\_peak-(i-1)

endif

if((cmpstr(Overlaps,"Separate")==0)&(Kill\_it==0))

ii=i

// Fall\_fit\_Extrap[ii-1]="1"

Do

Fall\_fit[ii]=""

// Wave/T Fall\_fit\_Extrap=Fall\_fit\_Extrap

// Fall\_fit\_Extrap[ii]=num2str(2+ii-i)

WaveStats/Q/R = (T\_Max[ii-1],T\_Max[ii]) Working\_trace\_copy

T\_Bkg1[ii]=V\_minloc

ii+=1

while(ii<=Last\_overlapping\_peak-1)

Calc\_Separated\_peak\_param(i-1,Last\_overlapping\_peak-1)

print "Overlapping spikes",i,"to",Last\_overlapping\_peak,"were Separated."

i=Last\_overlapping\_peak

endif

if((cmpstr(Overlaps,"Split")==0)&(Kill\_it==0))

T\_Bkg1[i-1,Last\_overlapping\_peak-1]=T\_Bkg1[i-1]

ii=i

Do

Fall\_fit[ii]=""

Fall\_fit[ii-1]=""

WaveStats/Q/R = (T\_Max[ii-1],T\_Max[ii]) Working\_trace\_copy

Peak\_Split2[ii-1]=V\_minloc

Peak\_Split1[ii]=V\_minloc

Calc\_split\_peak\_param(ii-1)

Calc\_split\_peak\_param(ii)

ii+=1

while(ii<Last\_overlapping\_peak)

print "Overlapping spikes",i,"to",Last\_overlapping\_peak,"were Split."

i=Last\_overlapping\_peak

endif

endif

i+=1

while(i<=Total\_peaks\_number)

Killwaves/Z Peak\_Delta

String Extrap\_

String Extrap\_waves=WaveList("Extrap\*",";","")

ii=0

Do

Extrap\_=StringFromList(ii,Extrap\_waves)

If(strlen(Extrap\_)==0)

break

endif

AppendToGraph/W=Main\_window $Extrap\_

ModifyGraph lstyle($Extrap\_)=2,lsize($Extrap\_)=0.5,rgb($Extrap\_)=(0,15872,65280)

ii+=1

while(1)

if(numpnts(Peak\_Num))

Wavestats/Q Peak\_Num

Total\_peaks\_number=V\_npnts

else

Total\_peaks\_number=0

endif

End

Function Find\_level\_plus\_2SD(StartX,LimitX,LevelY)

Variable StartX,LimitX,LevelY

SetDataFolder $"root:Quanta"

Wave Working\_trace\_copy=Working\_trace\_copy

NVAR Bkg\_noise\_I=Bkg\_noise\_I

NVAR Smoothing\_Factor=Smoothing\_Factor

Duplicate/O/R=(StartX,LimitX) Working\_trace\_copy, TEMP\_wave

Variable Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Smoothing\_Factor/2)

Smooth Binomial\_coeff, TEMP\_wave

FindLevel/Q/R=(StartX,LimitX) TEMP\_wave, Working\_trace\_copy(LevelY)

if (V\_LevelX)

FindLevel/Q/R=(V\_LevelX,LimitX) Working\_trace\_copy, Working\_trace\_copy(LevelY)

endif

if (V\_Flag)

FindLevel/Q/R=(StartX,LimitX) TEMP\_wave, (Working\_trace\_copy(LevelY)+Bkg\_noise\_I)

if (V\_Flag)

FindLevel/Q/R=(StartX,LimitX) TEMP\_wave, (Working\_trace\_copy(LevelY)+2\*Bkg\_noise\_I)

endif

endif

KillWaves/Z TEMP\_wave

if (V\_Flag)

Return 0

else

Return V\_LevelX

endif

End

Function Add\_Peak\_Manually(ctrlName) : ButtonControl

String ctrlname

SetDataFolder $"root:Quanta"

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Peak\_ID=Peak\_ID

Wave Peak\_Num=Peak\_Num

Wave T\_Max=T\_Max

Wave Working\_Trace\_Copy=Working\_Trace\_Copy

If(numpnts(Working\_trace\_copy)<2)

abort "No trace loaded or the wave is too short."

endif

Variable Start\_X=min(xcsr(A),xcsr(B))

Variable End\_X=max(xcsr(A),xcsr(B))

Variable New\_T\_Max=Find\_One\_Peak(Start\_X,End\_X)

If (New\_T\_Max==0)

abort

endif

Variable New\_peak\_pnt=New\_peak\_position(New\_T\_Max)

If (New\_peak\_pnt<0)

If(New\_peak\_pnt==-0.5)

New\_peak\_pnt=0

else

New\_peak\_pnt=abs(New\_peak\_pnt)

endif

If (Total\_peaks\_number!=1) // it is not the very first peak

Peak\_ID=New\_peak\_pnt+1

endif

abort "Spike with this time at maximum already exists!"

endif

If (T\_Max[Total\_peaks\_number-1]!=0) // it is not the very first peak

Change\_waves("Insert",New\_peak\_pnt)

endif

Peak\_Num[]=p+1

T\_Max[New\_peak\_pnt]=New\_T\_Max

Wavestats/Q Peak\_Num

Total\_peaks\_number=V\_npnts

FindLevel /Q/R=(T\_Max[New\_peak\_pnt],Start\_X ) Working\_Trace\_Copy, Working\_Trace\_Copy(Start\_X)

If(V\_flag==0)

Start\_X=V\_LevelX

endif

Calc\_Peak\_Parameters(New\_peak\_pnt,Start\_X,End\_X)

Peak\_ID=New\_peak\_pnt+1

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Main\_window

if (CheckName("Zoom\_Win", 6)!=0)

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Zoom\_Win

endif

Draw\_lines\_All(New\_peak\_pnt)

End

Function Find\_One\_Peak(Start\_X, End\_X)

Variable Start\_X

Variable End\_X

Variable Delta\_X=End\_X-Start\_X

SetDataFolder $"root:Quanta"

NVAR Detection\_Mult=Detection\_Mult

NVAR Smoothing\_Factor\_diff1=Smoothing\_Factor\_diff1

NVAR Bkg\_noise\_dI=Bkg\_noise\_dI

NVAR Smooth\_more=Smooth\_more

NVAR Smooth\_Derivative=Smooth\_Derivative

NVAR Smoothing\_Factor\_Add=Smoothing\_Factor\_Add

Wave diff1=diff1

Wave Working\_trace\_copy=Working\_trace\_copy

Bkg\_noise\_Calc()

Variable Binomial\_coeff

If (Smooth\_more==1)

If((x2pnt(Working\_trace\_copy,(End\_X+3\*Delta\_X))-x2pnt(Working\_trace\_copy,(Start\_X-3\*Delta\_X)))<Smoothing\_Factor\_Add)

DoAlert 0, "Smoothing of the trace requires more datapoints.\rReduce Additional Smoothing factor (Detection panel)."

Return 0

endif

Duplicate/O/R=((Start\_X-3\*Delta\_X),(End\_X+3\*Delta\_X)) Working\_trace\_copy, diff1, diff1\_nosmooth

Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Smoothing\_Factor\_Add)

Smooth Binomial\_coeff, diff1

else

Duplicate/O/R=((Start\_X-3\*Delta\_X),(End\_X+3\*Delta\_X)) Working\_trace\_copy, diff1, diff1\_nosmooth

endif

Differentiate diff1, diff1\_nosmooth

If (Smooth\_Derivative==1)

Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Smoothing\_Factor\_diff1)

Smooth Binomial\_coeff, diff1, diff1\_nosmooth

endif

Variable Detection\_level\_diff1=Bkg\_noise\_dI\*Detection\_Mult

Variable New\_T\_Max

wavestats/Q/R=(Start\_X,End\_X) Working\_trace\_copy

New\_T\_Max=V\_maxloc

FindPeak/Q/M=(Detection\_level\_diff1)/R=(Start\_X,End\_X) diff1

if (V\_Flag)

String Message="No events found between the coursors. \rTry reducing detection threshold and then repeate the detection.\rIf you want to add this spike anyway, press 'Yes'. "

DoAlert 1,Message

If (V\_Flag==1)

Return New\_T\_Max

else

Return 0

endif

else

FindLevel /Q/R=(V\_PeakLoc,xcsr(B)) diff1, 0

if (V\_Flag)

Message="This spike does not return to the baseline level. \rReduce the 'Smooth for 1st derrivative' factor under 'Options' menu.\rIf you want to add the spike anyway, press 'Yes'. "

DoAlert 1,Message

If (V\_Flag==1)

Return New\_T\_Max

else

Return 0

endif

else

Return New\_T\_Max

endif

endif

End

Function Split(ctrlName) : ButtonControl

String ctrlname

SetDataFolder $"root:Quanta"

NVAR Total\_peaks\_number=Total\_peaks\_number

If(Total\_peaks\_number==0)

abort

endif

Wave Peak\_Num=Peak\_Num

Wave T\_Max=T\_Max

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

NVAR Peak\_ID=Peak\_ID

Variable Peak\_pnt=Peak\_ID-1

NVAR Detection\_Mult=Detection\_Mult

NVAR Smoothing\_Factor\_diff1=Smoothing\_Factor\_diff1

Wave Working\_trace\_copy=Working\_trace\_copy

Variable L=max(T\_Bkg1[Peak\_pnt], Peak\_Split1[Peak\_pnt])

Variable R

If (Peak\_Split2[Peak\_pnt]!=0)

R=Peak\_Split2[Peak\_pnt]

else

R=T\_Bkg2[Peak\_pnt]

endif

If ((xcsr(A)<=L)%|(xcsr(A)>=R))

abort "Splitting point (round coursor) has to be between the Start and the End of the spike."

endif

Variable Split\_pnt=xcsr(A)

String Split\_Where

Variable Common\_bkg1=T\_Bkg1[Peak\_pnt]

Variable Common\_bkg2=T\_Bkg2[Peak\_pnt]

Variable Start\_X, End\_X

If (T\_Max[Peak\_pnt]<Split\_pnt)

If (Peak\_Split2[Peak\_pnt]!=0)

Split\_Where="Right\_Mid"

End\_X=Peak\_Split2[Peak\_pnt]

else

Split\_Where="Right\_Last"

End\_X=T\_Bkg2[Peak\_pnt]

endif

Start\_X=Split\_pnt

else

If (Peak\_Split1[Peak\_pnt]!=0)

Split\_Where="Left\_Mid"

Start\_X=Peak\_Split1[Peak\_pnt]

else

Split\_Where="Left\_First"

Start\_X=T\_Bkg1[Peak\_pnt]

endif

End\_X=Split\_pnt

endif

Variable New\_T\_Max

WaveStats /Q /R = (Start\_X, End\_X ) Working\_trace\_copy

New\_T\_Max=V\_maxloc

If(New\_T\_Max==0)

abort

endif

Variable New\_peak\_pnt=New\_peak\_position(New\_T\_Max)

If (New\_peak\_pnt<0)

If(New\_peak\_pnt==-0.5)

New\_peak\_pnt=0

else

New\_peak\_pnt=abs(New\_peak\_pnt)

endif

Peak\_ID=New\_peak\_pnt+1

abort "Spike with this time already exists!"

endif

If (T\_Max[Peak\_pnt]<Split\_pnt)

Peak\_Split2[Peak\_pnt]=Split\_pnt

else

Peak\_Split1[Peak\_pnt]=Split\_pnt

endif

Change\_waves("Insert",New\_peak\_pnt)

Peak\_Num[]=p+1

T\_Max[New\_peak\_pnt]=New\_T\_Max

T\_Bkg1[New\_peak\_pnt]=Common\_bkg1

T\_Bkg2[New\_peak\_pnt]=Common\_bkg2

If (strsearch(Split\_Where,"Right",0)!=-1)

Peak\_Split1[New\_peak\_pnt]=Split\_pnt

If (cmpstr(Split\_Where,"Right\_Mid")==0)

Peak\_Split2[New\_peak\_pnt]=Peak\_Split1[New\_peak\_pnt+1]

endif

endif

If (strsearch(Split\_Where,"Left",0)!=-1)

Peak\_Split2[New\_peak\_pnt]=Split\_pnt

If (cmpstr(Split\_Where,"Left\_Mid")==0)

Peak\_Split1[New\_peak\_pnt]=Peak\_Split2[New\_peak\_pnt-1]

endif

endif

// recalculate parameters for the original spike

If (strsearch(Split\_Where,"Right",0)!=-1)

Calc\_split\_peak\_param(New\_peak\_pnt-1)

else

Calc\_split\_peak\_param(New\_peak\_pnt+1)

endif

// calculate parameters for the new spike

Calc\_split\_peak\_param(New\_peak\_pnt)

Peak\_ID=New\_peak\_pnt+1

Wavestats/Q Peak\_Num

Total\_peaks\_number=V\_npnts

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Main\_window

if (CheckName("Zoom\_Win", 6)!=0)

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Zoom\_Win

endif

Draw\_lines\_All(New\_peak\_pnt)

End

Function Calc\_Split\_Peak\_Param(Peak\_pnt)

Variable Peak\_pnt

SetDataFolder $"root:Quanta"

Wave Peak\_Half\_H1=Peak\_Half\_H1

Wave Peak\_Half\_H2=Peak\_Half\_H2

Wave T\_Max=T\_Max

Wave Peak\_Base=Peak\_Base

Wave Peak\_Imax=Peak\_Imax

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Peak\_t05=Peak\_t05

Wave Peak\_Q=Peak\_Q

Wave Peak\_Molec=Peak\_Molec

Wave Working\_trace\_copy=Working\_trace\_copy

Variable Start\_X=T\_Bkg1[Peak\_pnt]

Variable End\_X=T\_Bkg2[Peak\_pnt]

Variable Max\_X=T\_Max[Peak\_pnt]

Variable Bkg\_under\_the\_Max=Working\_trace\_copy(Start\_X)+(Working\_trace\_copy(End\_X)-Working\_trace\_copy(Start\_X))\*(Max\_X-Start\_X)/(End\_X-Start\_X)

Variable Half\_Height=Working\_trace\_copy(Max\_X) - (Working\_trace\_copy(Max\_X)-Bkg\_under\_the\_Max)/2

Peak\_Imax[Peak\_pnt]=Working\_trace\_copy(Max\_X)-Bkg\_under\_the\_Max

Start\_X=max(T\_Bkg1[Peak\_pnt], Peak\_Split1[Peak\_pnt])

FindLevel /Q/R=(Start\_X,Max\_X) Working\_trace\_copy, Half\_Height

If(V\_Flag==1)

Peak\_Half\_H1[Peak\_pnt]=Peak\_Split1[Peak\_pnt]

else

Peak\_Half\_H1[Peak\_pnt]=min(V\_LevelX,Max\_X)

endif

If (Peak\_Split2[Peak\_pnt]!=0)

End\_X=Peak\_Split2[Peak\_pnt]

else

End\_X=T\_Bkg2[Peak\_pnt]

endif

FindLevel /Q/R=(Max\_X, End\_X) Working\_trace\_copy, Half\_Height

If (V\_flag==1)

Peak\_Half\_H2[Peak\_pnt]=End\_X

else

Peak\_Half\_H2[Peak\_pnt]=max(V\_LevelX,Max\_X)

endif

Peak\_Base[Peak\_pnt]=(End\_X-Start\_X)\*1000

Peak\_t05[Peak\_pnt]=(Peak\_Half\_H2[Peak\_pnt]-Peak\_Half\_H1[Peak\_pnt])\*1000

Variable Area\_total=area(Working\_trace\_copy,Start\_X,End\_X)

Variable Bkg1\_Y, Bkg2\_Y

If (Peak\_Split1[Peak\_pnt]!=0)

Bkg1\_Y=Y\_offset(T\_Bkg1[Peak\_pnt], T\_Bkg2[Peak\_pnt], Peak\_Split1[Peak\_pnt])

else

Bkg1\_Y=Working\_trace\_copy(T\_Bkg1[Peak\_pnt])

endif

If (Peak\_Split2[Peak\_pnt]!=0)

Bkg2\_Y=Y\_offset(T\_Bkg1[Peak\_pnt], T\_Bkg2[Peak\_pnt], Peak\_Split2[Peak\_pnt])

else

Bkg2\_Y=Working\_trace\_copy(T\_Bkg2[Peak\_pnt])

endif

Variable Area\_bkg=(End\_X-Start\_X)\*(Bkg1\_Y+Bkg2\_Y)/2

Peak\_Q[Peak\_pnt]=Area\_total-Area\_bkg

Peak\_Molec[Peak\_pnt]=(Peak\_Q[Peak\_pnt])\*3.121\*10^6

Calc\_Rise\_Fall(Peak\_pnt)

End

Function/S Check\_for\_Separated\_peaks(Peak\_pnt)

Variable Peak\_pnt

SetDataFolder $"root:Quanta"

Wave T\_Max=T\_Max

Wave Peak\_Imax=Peak\_Imax

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Working\_trace\_copy=Working\_trace\_copy

NVAR Total\_peaks\_number=Total\_peaks\_number

Variable Start\_X=T\_Bkg1[Peak\_pnt]

Variable End\_X=T\_Bkg2[Peak\_pnt]

Variable Max\_X=T\_Max[Peak\_pnt]

Variable First\_Overlapp\_peak=0,Last\_Overlapp\_peak=0

If ((Peak\_Split2[Peak\_pnt]!=0)%|(Peak\_Split1[Peak\_pnt]!=0))

return "0;0"

endif

If ((T\_Max[Peak\_pnt]!=T\_Max[Peak\_pnt+1])&(T\_Max[Peak\_pnt+1])<(T\_Bkg2[Peak\_pnt]))

First\_Overlapp\_peak=Peak\_pnt

endif

Variable i=Peak\_pnt-1

Do

If ((T\_Max[Peak\_pnt]!=T\_Max[i])&(T\_Max[Peak\_pnt])<(T\_Bkg2[i]))

First\_Overlapp\_peak=i

else

break

endif

i-=1

while (i>=0)

Variable ii

ii=First\_Overlapp\_peak+1

Do

If (T\_Max[ii]<T\_Bkg2[First\_Overlapp\_peak])

Last\_Overlapp\_peak=ii

else

break

endif

ii+=1

while (ii<=Total\_peaks\_number-1)

if(Total\_peaks\_number<2)

return "0;0"

endif

If((First\_Overlapp\_peak!=Last\_Overlapp\_peak)&(T\_Bkg2[First\_Overlapp\_peak]>T\_Max[Peak\_pnt]))

String First\_and\_Last=num2str(First\_Overlapp\_peak)+";"+num2str(Last\_Overlapp\_peak)

return First\_and\_Last

else

return "0;0"

endif

End

Function Calc\_Separated\_Peak\_Param(First\_Overlapp\_pnt,Last\_Overlapp\_pnt)

Variable First\_Overlapp\_pnt

Variable Last\_Overlapp\_pnt

SetDataFolder $"root:Quanta"

Wave Peak\_Half\_H1=Peak\_Half\_H1

Wave Peak\_Half\_H2=Peak\_Half\_H2

Wave T\_Max=T\_Max

Wave Peak\_Base=Peak\_Base

Wave Peak\_Imax=Peak\_Imax

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Peak\_t05=Peak\_t05

Wave Peak\_Q=Peak\_Q

Wave Peak\_Molec=Peak\_Molec

Wave Fall\_slope=Fall\_slope

Wave Fall\_slope2=Fall\_slope2

Wave Working\_trace\_copy=Working\_trace\_copy

Wave/T Fall\_Fit=Fall\_Fit

NVAR Total\_peaks\_number=Total\_peaks\_number

SVAR Fit\_method=Fit\_method

Wave/T Fall\_fit\_Extrap=Fall\_fit\_Extrap

String Extrapolation\_name=""

Variable A0, A1, tau1,Slope

String Extrap\_Exists

Variable Half\_Height, Max\_X, Start\_X, End\_X, End\_Common

Variable Area\_total, Area\_bkg, Area\_Extra=0

End\_Common=T\_Bkg2[First\_Overlapp\_pnt]

Variable i=First\_Overlapp\_pnt // First peak pnt

Do

If(cmpstr(Fall\_fit\_Extrap[i],"")==0)

Fall\_fit\_Extrap[i]="N:;Total:;Fit:;Tau:"

endif

If(cmpstr(StringByKey("N", Fall\_fit\_Extrap[i]),"")==0)

Fall\_fit\_Extrap[i] = ReplaceStringByKey("N", Fall\_fit\_Extrap[i], num2str(1+i-First\_Overlapp\_pnt))

Fall\_fit\_Extrap[i] = ReplaceStringByKey("Total", Fall\_fit\_Extrap[i], num2str(1+Last\_Overlapp\_pnt-First\_Overlapp\_pnt))

endif

Max\_X=T\_Max[i]

Start\_X=T\_Bkg1[i]

T\_Bkg2[i]=End\_Common

If(i==Last\_Overlapp\_pnt)

End\_X=T\_Bkg2[i]

else

End\_X=T\_Bkg1[i+1]

endif

If(i==First\_Overlapp\_pnt)

Variable Bkg\_under\_the\_Max=Working\_trace\_copy(T\_Bkg1[i])

Bkg\_under\_the\_Max=Bkg\_under\_the\_Max+(Working\_trace\_copy(T\_Bkg2[i])-Working\_trace\_copy(T\_Bkg1[i]))\*(T\_Max[i]-T\_Bkg1[i])/(T\_Bkg2[i]-T\_Bkg1[i])

Peak\_Imax[i]=Working\_trace\_copy(T\_Max[i])-Bkg\_under\_the\_Max

else

Extrapolation\_name="Extrap\_"+num2str(i) // extrap curve for the previous peak

Wave Extrap\_prev=$Extrapolation\_name

Peak\_Imax[i]=Working\_trace\_copy(Max\_X)-Extrap\_prev(Max\_X)

endif

Half\_Height=Working\_trace\_copy(Max\_X)-Peak\_Imax[i]/2

FindLevel /Q/R=(Max\_X,Start\_X) Working\_trace\_copy, Half\_Height

Peak\_Half\_H1[i]=min(V\_LevelX,Max\_X)

If(i!=First\_Overlapp\_pnt)

FindLevel /Q/R=(Max\_X,Start\_X) Extrap\_prev, Half\_Height

if(V\_Flag==0)

Peak\_Half\_H1[i]=max(V\_LevelX,Peak\_Half\_H1[i])

endif

endif

Calc\_Rise\_Fall(i)

Extrapolation\_name="Extrap\_"+num2str(i+1)

If(exists(Extrapolation\_name)==1)

if (CheckName("Zoom\_Win", 6)!=0)

RemoveFromGraph/Z/W=Zoom\_Win $Extrapolation\_name

endif

RemoveFromGraph/Z/W=Main\_window $Extrapolation\_name

KillWaves /Z $Extrapolation\_name

endif

If(i!=Last\_Overlapp\_pnt)

Duplicate/O/R=(T\_Bkg1[i+1],T\_Bkg2[i]) Working\_trace\_copy qqq

SetScale/I x 0,(T\_Bkg2[i]-T\_Bkg1[i+1]),"s", qqq

A1=Working\_trace\_copy(T\_Bkg1[i+1])-Working\_trace\_copy(T\_Bkg2[i])

A0=Working\_trace\_copy(T\_Bkg2[i])

String FitFunction=StringByKey("Fit", Fall\_fit\_Extrap[i])

If(strlen(FitFunction)==0)

FitFunction=Fall\_Fit[i]

Fall\_fit\_Extrap[i] = ReplaceStringByKey("Fit", Fall\_fit\_Extrap[i], FitFunction)

endif

If((A1>0)&(cmpstr(FitFunction,"Line")!=0))

String ExpTau=StringByKey("Tau", Fall\_fit\_Extrap[i])

If(strlen(ExpTau)!=0)

tau1=str2num(ExpTau)

else

tau1=max(Fall\_slope[i],Fall\_slope2[i])

endif

Fall\_fit\_Extrap[i] = ReplaceStringByKey("Tau", Fall\_fit\_Extrap[i], num2str(tau1))

tau1=1/tau1\*1000

qqq=A0+A1\*exp(-x\*tau1)

SetScale/I x T\_Bkg1[i+1],T\_Bkg2[i],"s", qqq

Duplicate/O qqq $Extrapolation\_name

else

Slope=A1/(T\_Bkg2[i]-T\_Bkg1[i+1])

qqq=Working\_trace\_copy(T\_Bkg1[i+1])-Slope\*x

SetScale/I x T\_Bkg1[i+1],T\_Bkg2[i],"s", qqq

Duplicate/O qqq $Extrapolation\_name

// If(cmpstr(StringByKey("Fit", Fall\_fit\_Extrap[i]),"")==0)

Fall\_fit\_Extrap[i] = ReplaceStringByKey("Fit", Fall\_fit\_Extrap[i], "Line")

Fall\_fit\_Extrap[i] = ReplaceStringByKey("Tau", Fall\_fit\_Extrap[i], num2str(-1))

// endif

endif

KillWaves /Z qqq

endif

FindLevel /Q/R=(Max\_X, End\_X) Working\_trace\_copy, Half\_Height

If (V\_flag==1)

If(i==Last\_Overlapp\_pnt)

Peak\_Half\_H2[i]=End\_X

else

Extrapolation\_name="Extrap\_"+num2str(i+1) // extrap curve for the current peak

Wave Extrap\_current=$Extrapolation\_name

FindLevel /Q/R=(T\_Bkg1[i+1], T\_Bkg2[i]) Extrap\_current, Half\_Height

If (V\_flag==0)

Peak\_Half\_H2[i]=V\_LevelX

else

Peak\_Half\_H2[i]=End\_X

endif

endif

else

Peak\_Half\_H2[i]=max(V\_LevelX,Max\_X)

endif

Peak\_Base[i]=(T\_Bkg2[i]-T\_Bkg1[i])\*1000

Peak\_t05[i]=(Peak\_Half\_H2[i]-Peak\_Half\_H1[i])\*1000

i+=1

while (i<=Last\_Overlapp\_pnt)

i=Last\_Overlapp\_pnt // Last peak pnt

Do

Area\_total=area(Working\_trace\_copy,T\_Bkg1[i],T\_Bkg2[i])

If(i==First\_Overlapp\_pnt)

Area\_bkg=(T\_Bkg2[i]-T\_Bkg1[i])\*(Working\_trace\_copy(T\_Bkg1[i])+Working\_trace\_copy(T\_Bkg2[i]))/2

Peak\_Q[i]=Area\_total-Area\_bkg-Area\_Extra

else

Extrapolation\_name="Extrap\_"+num2str(i) // extrap curve for the previous peak

Wave Extrap\_prev=$Extrapolation\_name

Area\_bkg=area(Extrap\_prev,T\_Bkg1[i],T\_Bkg2[i])

Peak\_Q[i]=Area\_total-Area\_bkg-Area\_Extra

Area\_Extra+=Peak\_Q[i]

endif

Peak\_Molec[i]=(Peak\_Q[i])\*3.121\*10^6

i-=1

while (i>=First\_Overlapp\_pnt)

End

Function New\_Peak\_Position(New\_T\_Max)

Variable New\_T\_Max

SetDataFolder $"root:Quanta"

Wave T\_Max=T\_Max

NVAR Total\_peaks\_number=Total\_peaks\_number

If (Total\_peaks\_number==1)

If (T\_Max[Total\_peaks\_number-1]==0) // no peaks exists yet

Change\_waves("Make",0)

return 0

endif

endif

Variable Delta=CmpNum((T\_Max[Total\_peaks\_number-1]),New\_T\_Max,18)

If (Delta==0) // peak already exists

If (Total\_peaks\_number==1) // and an existing peak is the only one

return -1

else

return (-(Total\_peaks\_number-1))

endif

endif

If (Delta==-1)

return (Total\_peaks\_number) // new peak will be the last one

endif

Variable Peak\_pnt=0

Do

Delta=CmpNum((T\_Max[Peak\_pnt]),New\_T\_Max,18)

If (Delta>=0)

If (Delta==0) // peak already exists

if (Peak\_pnt==0)

return (-0.5)

else

return (-Peak\_pnt)

endif

else

return (Peak\_pnt)

endif

endif

Peak\_pnt+=1

while(Total\_peaks\_number)

End

Function Calc\_Peak\_Parameters(Peak\_pnt,Start\_X,End\_X)

Variable Peak\_pnt

Variable Start\_X

Variable End\_X

Variable Max\_X

SetDataFolder $"root:Quanta"

Wave T\_Max=T\_Max

Wave Peak\_Imax=Peak\_Imax

Wave Peak\_t05=Peak\_t05

Wave Peak\_Q=Peak\_Q

Wave Peak\_Base=Peak\_Base

Wave Peak\_Half\_H1=Peak\_Half\_H1

Wave Peak\_Half\_H2=Peak\_Half\_H2

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Working\_trace\_copy=Working\_trace\_copy

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave/T Fall\_fit\_Extrap=Fall\_fit\_Extrap

T\_Bkg1[Peak\_pnt]=Start\_X

T\_Bkg2[Peak\_pnt]=End\_X

// check for split overlapping spikes

If((Peak\_Split1[Peak\_pnt]!=0)%|(Peak\_Split2[Peak\_pnt]!=0))

Calc\_split\_peak\_param(Peak\_pnt)

return 0

endif

// check for separated overlapping spikes

String First\_and\_Last=Check\_for\_Separated\_peaks(Peak\_pnt)

Variable First\_separated\_pnt=str2num(StringFromList(0,First\_and\_Last))

Variable Last\_separated\_pnt=str2num(StringFromList(1,First\_and\_Last))

If(First\_separated\_pnt!=Last\_separated\_pnt)

Variable i=First\_separated\_pnt

Do

Fall\_fit\_Extrap[i]=""

i+=1

while(i<=Last\_separated\_pnt)

Calc\_Separated\_peak\_param(First\_separated\_pnt,Last\_separated\_pnt)

return 0

endif

// find Imax, pA

Max\_X=T\_Max[Peak\_pnt]

Variable Bkg\_under\_the\_Max=Working\_trace\_copy(T\_Bkg1[Peak\_pnt])+(Working\_trace\_copy(T\_Bkg2[Peak\_pnt])-Working\_trace\_copy(T\_Bkg1[Peak\_pnt]))\*(Max\_X-T\_Bkg1[Peak\_pnt])/(T\_Bkg2[Peak\_pnt]-T\_Bkg1[Peak\_pnt])

Peak\_Imax[Peak\_pnt]=Working\_trace\_copy(Max\_X)-Bkg\_under\_the\_Max

// find t(1/2), ms

Variable Half\_Height=Working\_trace\_copy(Max\_X) - (Working\_trace\_copy(Max\_X)-Bkg\_under\_the\_Max)/2

FindLevel /Q/R=(Max\_X,Start\_X) Working\_trace\_copy, Half\_Height

Peak\_Half\_H1[Peak\_pnt]=min(V\_LevelX,Max\_X)

FindLevel /Q/R=(Max\_X, End\_X) Working\_trace\_copy, Half\_Height

If (V\_flag==1)

Peak\_Half\_H2[Peak\_pnt]=End\_X

else

Peak\_Half\_H2[Peak\_pnt]=max(V\_LevelX,Max\_X)

endif

Peak\_t05[Peak\_pnt]=(Peak\_Half\_H2[Peak\_pnt]-Peak\_Half\_H1[Peak\_pnt])\*1000

Peak\_Base[Peak\_pnt]=(T\_Bkg2[Peak\_pnt]-T\_Bkg1[Peak\_pnt])\*1000

// find Q, pC, molecules

Calc\_Peak\_Q(Peak\_pnt,T\_Bkg1[Peak\_pnt],T\_Bkg2[Peak\_pnt])

// find rise and fall parameters

If ((Peak\_Q[Peak\_pnt]>0)%|(Peak\_t05[Peak\_pnt]>0))

Calc\_Rise\_Fall(Peak\_pnt)

endif

// If ((Detection\_limits(Peak\_pnt)!=1))

// Calc\_Rise\_Fall(Peak\_pnt)

// endif

End

Function Fall\_fit\_PopMenu(theTag,popNum,popStr) : PopupMenuControl

String theTag

Variable popNum

String popStr

SetDataFolder $"root:Quanta"

SVAR Fit\_method=Fit\_method

NVAR Peak\_ID=Peak\_ID

// Wave/T Fall\_fit=Fall\_fit

Variable Current\_Peak=Peak\_ID

If(cmpstr(popStr,Fit\_method)!=0)

Fit\_method=popStr

Change\_Table ("q",0)

If(cmpstr(Fit\_method,"DblExp")==0)

SetVariable Results\_Fall\_Chi, win=Options\_Tab\_Panels, disable=0

else

SetVariable Results\_Fall\_Chi, win=Options\_Tab\_Panels, disable=1

endif

endif

End

Function Fall\_Extrap\_PopMenu\_Single(theTag,popNum,popStr) : PopupMenuControl

String theTag

Variable popNum

String popStr

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

Wave/T Fall\_fit\_Extrap=Fall\_fit\_Extrap

Variable Peak\_pnt=Peak\_ID-1

String First\_and\_Last=Check\_for\_Separated\_peaks(Peak\_pnt)

Variable First\_separated\_pnt=str2num(StringFromList(0,First\_and\_Last))

Variable Last\_separated\_pnt=str2num(StringFromList(1,First\_and\_Last))

Fall\_fit\_Extrap[Peak\_pnt] = ReplaceStringByKey("Fit", Fall\_fit\_Extrap[Peak\_pnt], popStr)

Fall\_fit\_Extrap[Peak\_pnt] = ReplaceStringByKey("Tau", Fall\_fit\_Extrap[Peak\_pnt], "")

Calc\_Separated\_peak\_param(First\_separated\_pnt,Last\_separated\_pnt)

Draw\_lines\_All(Peak\_pnt)

End

Function Fall\_fit\_PopMenu\_Single(theTag,popNum,popStr) : PopupMenuControl

String theTag

Variable popNum

String popStr

SetDataFolder $"root:Quanta"

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Peak\_ID=Peak\_ID

Wave Fall\_ChiRatio=Fall\_ChiRatio

Wave Fall\_time=Fall\_time

Wave Fall\_slope=Fall\_slope

Wave Fall\_slope2=Fall\_slope2

Wave/T Fall\_fit=Fall\_fit

Wave/T Fall\_fit\_Extrap=Fall\_fit\_Extrap

Variable Peak\_pnt=Peak\_ID-1

String Fit\_Function=Fall\_fit[Peak\_pnt]

If((cmpstr(popStr,Fit\_Function)!=0)&(Total\_peaks\_number!=0))

Fall\_fit[Peak\_pnt]=popStr

String Slope\_coeffs=Fit\_Fall\_Decay(Peak\_pnt,popStr)

If(cmpstr(popStr,"DblExp")==0)

String Slope\_coeffs\_Exp=Fit\_Fall\_Decay(Peak\_pnt,"Exp")

Variable Ratio=str2num(StringFromList(3,Slope\_coeffs\_Exp))/str2num(StringFromList(3,Slope\_coeffs))

Fall\_ChiRatio[Peak\_pnt]=Ratio

else

Fall\_ChiRatio[Peak\_pnt]=1

endif

Fall\_time[Peak\_pnt]=str2num(StringFromList(0,Slope\_coeffs))

Fall\_slope[Peak\_pnt]=str2num(StringFromList(1,Slope\_coeffs))

Fall\_slope2[Peak\_pnt]=str2num(StringFromList(2,Slope\_coeffs))

String First\_and\_Last=Check\_for\_Separated\_peaks(Peak\_pnt)

Variable First\_separated\_pnt=str2num(StringFromList(0,First\_and\_Last))

Variable Last\_separated\_pnt=str2num(StringFromList(1,First\_and\_Last))

If(First\_separated\_pnt!=Last\_separated\_pnt)

Fall\_fit\_Extrap[Peak\_pnt] = ReplaceStringByKey("Fit", Fall\_fit\_Extrap[Peak\_pnt], "")

Fall\_fit\_Extrap[Peak\_pnt] = ReplaceStringByKey("Tau", Fall\_fit\_Extrap[Peak\_pnt], "")

Calc\_Separated\_peak\_param(First\_separated\_pnt,Last\_separated\_pnt)

endif

Draw\_lines\_All(Peak\_pnt)

endif

End

Function Calc\_Rise\_Fall(Peak\_pnt)

Variable Peak\_pnt

SetDataFolder $"root:Quanta"

Wave T\_Max=T\_Max

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Peak\_Half\_H1=Peak\_Half\_H1

Wave Peak\_Imax=Peak\_Imax

Wave Rise\_time=Rise\_time

Wave Rise\_Midpoint=Rise\_Midpoint

Wave Rise\_Lowpnt\_X=Rise\_Lowpnt\_X

Wave Rise\_Hipnt\_X=Rise\_Hipnt\_X

Wave Rise\_slope=Rise\_slope

Wave Fall\_time=Fall\_time

Wave Fall\_slope=Fall\_slope

Wave Fall\_slope2=Fall\_slope2

Wave Fall\_ChiRatio=Fall\_ChiRatio

Wave Working\_trace\_copy=Working\_trace\_copy

Wave Foot\_W=Foot\_W

Wave Foot\_end=Foot\_end

Wave diff1\_nosmooth=diff1\_nosmooth

NVAR Rise\_Low\_Prc=Rise\_Low\_Prc

NVAR Rise\_Hi\_Prc=Rise\_Hi\_Prc

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Fall\_ChiRatio\_Cutoff=Fall\_ChiRatio\_Cutoff

SVAR Fit\_method=Fit\_method

Variable Start\_X=T\_Bkg1[Peak\_pnt]

Variable End\_X=T\_Bkg2[Peak\_pnt]

Variable dI\_max\_Y,dI\_max\_X

Variable Half\_Hight\_L=(Working\_trace\_copy(T\_Max[Peak\_pnt])-Working\_trace\_copy(Start\_X))/2

FindLevel/Q/R=(T\_Max[Peak\_pnt],Start\_X) Working\_trace\_copy, (Working\_trace\_copy(Start\_X)+Half\_Hight\_L )

Variable Half\_Hight\_L\_X=V\_LevelX

// Check for overlaps

If ((T\_Max[Peak\_pnt-1]!=T\_Max[Peak\_pnt])&(T\_Max[Peak\_pnt-1])>(T\_Bkg1[Peak\_pnt]))

Start\_X=T\_Bkg2[Peak\_pnt-1]

endif

If ((T\_Max[Peak\_pnt+1]!=T\_Max[Peak\_pnt])&(T\_Max[Peak\_pnt+1])<(T\_Bkg2[Peak\_pnt]))

End\_X=T\_Bkg1[Peak\_pnt+1]

endif

If(Peak\_Split1[Peak\_pnt])

Start\_X=Peak\_Split1[Peak\_pnt]

endif

If(Peak\_Split2[Peak\_pnt])

End\_X=Peak\_Split2[Peak\_pnt]

endif

If(Rise\_Midpoint[Peak\_pnt]==0)

If((pnt2x(diff1\_nosmooth,0)>Start\_X)%|(pnt2x(diff1\_nosmooth,numpnts(diff1\_nosmooth))<End\_X))

Duplicate/O Working\_trace\_copy, diff1\_nosmooth

Differentiate diff1\_nosmooth

NVAR Smoothing\_Factor\_diff1=Smoothing\_Factor\_diff1

Variable Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Smoothing\_Factor\_diff1)

Smooth/E=1 Binomial\_coeff, diff1\_nosmooth

endif

wavestats/Q/R=(T\_Max[Peak\_pnt],Half\_Hight\_L\_X), diff1\_nosmooth

Rise\_Midpoint[Peak\_pnt]=V\_maxloc

dI\_max\_X=V\_maxloc

dI\_max\_Y=max(Working\_trace\_copy(Start\_X)+Half\_Hight\_L, Working\_trace\_copy(dI\_max\_X))

else

dI\_max\_X=Rise\_Midpoint[Peak\_pnt]

dI\_max\_Y=Working\_trace\_copy(Rise\_Midpoint[Peak\_pnt])

endif

// calculate the upper and lower points for the linear fit

Variable Rise\_Delta=(Working\_trace\_copy(T\_Max[Peak\_pnt])-dI\_max\_Y)/2

Variable Three\_fouth\_Height\_L\_Y=dI\_max\_Y+Rise\_Delta

Variable One\_fouth\_Height\_L\_Y=dI\_max\_Y-Rise\_Delta

Variable One\_fouth\_Height\_L\_X, Three\_fouth\_Height\_L\_X

FindLevel/P/Q/R=(T\_Max[Peak\_pnt],Start\_X) Working\_trace\_copy, Three\_fouth\_Height\_L\_Y

Three\_fouth\_Height\_L\_X=pnt2x(Working\_trace\_copy, V\_LevelX )

Rise\_Hipnt\_X[Peak\_pnt]=Three\_fouth\_Height\_L\_X

FindLevel/P/Q/R=(T\_Max[Peak\_pnt],Start\_X) Working\_trace\_copy, One\_fouth\_Height\_L\_Y

One\_fouth\_Height\_L\_X=pnt2x(Working\_trace\_copy, V\_LevelX )

Rise\_Lowpnt\_X[Peak\_pnt]=One\_fouth\_Height\_L\_X

// find Rise Slope

If(Rise\_Lowpnt\_X[Peak\_pnt]==Rise\_Hipnt\_X[Peak\_pnt])

return 0

else

Variable Rise\_B\_coeff=Fit\_Rise(Rise\_Lowpnt\_X[Peak\_pnt],Rise\_Hipnt\_X[Peak\_pnt])

endif

Rise\_slope[Peak\_pnt]=Rise\_B\_coeff/1000

// find Foot end at the extrapolation of the linear fit to spike baseline

Foot\_End[Peak\_pnt]=Extrapolate\_to\_Baseline(Peak\_pnt)

// find Risetime

Variable Rise\_Segment=(Rise\_Hi\_Prc-Rise\_Low\_Prc)/100

Rise\_time[Peak\_pnt]=(T\_max[Peak\_pnt]-Foot\_end[Peak\_pnt])\*1000\*Rise\_Segment

// find foot width, hight and charge

Calc\_Foot\_Parameters(Peak\_pnt)

// recalculate rising phase parameters for spikes without the feet

if(Foot\_W[Peak\_pnt]==0)

Rise\_Midpoint[Peak\_pnt]=Peak\_Half\_H1[Peak\_pnt]

Variable Hight\_L=Working\_trace\_copy(T\_Max[Peak\_pnt])-Working\_trace\_copy(T\_Bkg1[Peak\_pnt])

Variable Upper\_Y=Working\_trace\_copy(T\_Bkg1[Peak\_pnt])+Hight\_L\*Rise\_Hi\_Prc/100

Variable Lower\_Y=Working\_trace\_copy(T\_Bkg1[Peak\_pnt])+Hight\_L\*Rise\_Low\_Prc/100

FindLevel/P/Q/R=(T\_Max[Peak\_pnt],T\_Bkg1[Peak\_pnt]) Working\_trace\_copy, Upper\_Y

Rise\_Hipnt\_X[Peak\_pnt]=pnt2x(Working\_trace\_copy, V\_LevelX )

FindLevel/P/Q/R=(T\_Max[Peak\_pnt],T\_Bkg1[Peak\_pnt]) Working\_trace\_copy, Lower\_Y

Rise\_Lowpnt\_X[Peak\_pnt]=pnt2x(Working\_trace\_copy, V\_LevelX )

Rise\_B\_coeff=Fit\_Rise(Rise\_Lowpnt\_X[Peak\_pnt],Rise\_Hipnt\_X[Peak\_pnt])

Rise\_slope[Peak\_pnt]=Rise\_B\_coeff/1000

Rise\_time[Peak\_pnt]=(T\_max[Peak\_pnt]-Foot\_end[Peak\_pnt])\*1000\*Rise\_Segment

endif

// find falling phase parameters

Wave/t Fall\_fit=Fall\_fit

Variable Change\_Fit=0

If(cmpstr(Fall\_fit[Peak\_pnt],"")==0)

Fall\_fit[Peak\_pnt]=Fit\_method

Change\_Fit=1

endif

String Fit\_Function=Fall\_fit[Peak\_pnt]

String Slope\_coeffs=Fit\_Fall\_Decay(Peak\_pnt,Fit\_Function)

If(cmpstr(Fit\_method,"DblExp")==0)

String Slope\_coeffs\_Exp=Fit\_Fall\_Decay(Peak\_pnt,"Exp")

Variable Ratio=str2num(StringFromList(3,Slope\_coeffs\_Exp))/str2num(StringFromList(3,Slope\_coeffs))

If(Ratio<Fall\_ChiRatio\_Cutoff)

if(Change\_Fit==1)

Slope\_coeffs=Slope\_coeffs\_Exp

Fall\_fit[Peak\_pnt]="Exp"

endif

endif

Fall\_ChiRatio[Peak\_pnt]=Ratio

else

Fall\_ChiRatio[Peak\_pnt]=1

endif

Fall\_time[Peak\_pnt]=str2num(StringFromList(0,Slope\_coeffs))

Fall\_slope[Peak\_pnt]=str2num(StringFromList(1,Slope\_coeffs))

Fall\_slope2[Peak\_pnt]=str2num(StringFromList(2,Slope\_coeffs))

End

Function Fit\_Rise(Start\_X,End\_X)

Variable Start\_X

Variable End\_X

SetDataFolder $"root:Quanta"

Make/O/N=0 W\_coef

If(x2pnt(Working\_trace\_copy,End\_X)-x2pnt(Working\_trace\_copy,Start\_X)<2)

return 0

endif

Duplicate/O/R=(Start\_X,End\_X) Working\_trace\_copy, Rise\_phase,qqq

SetScale/I x 0,(End\_X-Start\_X),"s", Rise\_phase

CurveFit/Q/N line Rise\_phase /D=Rise\_phase

SetScale/I x Start\_X,End\_X,"s", Rise\_phase

KillWaves /Z qqq

return W\_coef[1]

End

Function Extrapolate\_to\_Baseline(Peak\_pnt)

Variable Peak\_pnt

Wave T\_Bkg1=T\_Bkg1

Wave Rise\_Midpoint=Rise\_Midpoint

Wave Rise\_slope=Rise\_slope

Wave Working\_trace\_copy=Working\_trace\_copy

Variable Midpoint\_Y=Working\_trace\_copy(Rise\_Midpoint[Peak\_pnt])

Variable T\_Bkg1\_Y=Working\_trace\_copy(T\_Bkg1[Peak\_pnt])

Variable Rise\_B\_Coeff=Rise\_slope[Peak\_pnt]\*1000

Variable Foot\_X=(1/Rise\_B\_Coeff)\*(Midpoint\_Y-T\_Bkg1\_Y)

Foot\_X=(Rise\_Midpoint[Peak\_pnt]-Foot\_X)

Return Foot\_X

End

Function New\_Rise(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

Wave T\_Max=T\_Max

Wave T\_Bkg1=T\_Bkg1

Wave Foot\_End=Foot\_End

Wave Rise\_Lowpnt\_X=Rise\_Lowpnt\_X

Wave Rise\_Hipnt\_X=Rise\_Hipnt\_X

Wave Rise\_time=Rise\_time

Wave Rise\_Midpoint=Rise\_Midpoint

Wave Rise\_slope=Rise\_slope

Wave Working\_trace\_copy=Working\_trace\_copy

NVAR Rise\_Low\_Prc=Rise\_Low\_Prc

NVAR Rise\_Hi\_Prc=Rise\_Hi\_Prc

Variable Peak\_pnt=Peak\_ID-1

Make/O/N=0 W\_coef

Variable Start\_X, End\_X

If (xcsr(A)<xcsr(B))

Start\_X=xcsr(A)

End\_X=xcsr(B)

else

Start\_X=xcsr(B)

End\_X=xcsr(A)

endif

If ((Start\_X>=T\_Max[Peak\_pnt])%|(End\_X<=T\_Bkg1[Peak\_pnt]))

abort "The coursors have to be between spike's Start and Maximum!"

endif

Variable Rise\_B\_coeff=Fit\_Rise(Start\_X,End\_X)

Rise\_slope[Peak\_pnt]=Rise\_B\_coeff/1000

Rise\_Lowpnt\_X[Peak\_pnt]=Start\_X

Rise\_Hipnt\_X[Peak\_pnt]=End\_X

Rise\_Midpoint[Peak\_pnt]=pnt2x(Rise\_phase, round(numpnts(Rise\_phase)/2))

Foot\_End[Peak\_pnt]=Extrapolate\_to\_Baseline(Peak\_pnt)

Variable Rise\_Segment=(Rise\_Hi\_Prc-Rise\_Low\_Prc)/100

Rise\_time[Peak\_pnt]=(T\_max[Peak\_pnt]-Foot\_end[Peak\_pnt])\*1000\*Rise\_Segment

Calc\_Foot\_Parameters(Peak\_pnt)

Draw\_lines\_All(Peak\_pnt)

End

Function/S Fit\_Fall\_Decay(Peak\_pnt,Fit\_Function)

Variable Peak\_pnt

String Fit\_Function

SetDataFolder $"root:Quanta"

Wave T\_Max=T\_Max

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Fall\_time=Fall\_time

Wave Fall\_slope=Fall\_slope

Wave Fall\_slope2=Fall\_slope2

Wave Working\_trace\_copy=Working\_trace\_copy

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Fall\_ChiRatio\_Cutoff=Fall\_ChiRatio\_Cutoff

Make/O/N=0 W\_coef

Variable End\_X=T\_Bkg2[Peak\_pnt]

If ((T\_Max[Peak\_pnt+1]!=T\_Max[Peak\_pnt])&(T\_Max[Peak\_pnt+1])<(T\_Bkg2[Peak\_pnt]))

End\_X=T\_Bkg1[Peak\_pnt+1]

endif

If(Peak\_Split2[Peak\_pnt])

End\_X=Peak\_Split2[Peak\_pnt]

endif

Variable Hight\_R=Working\_trace\_copy(T\_Max[Peak\_pnt])-Working\_trace\_copy(End\_X)

Variable One\_fouth\_Height\_Y\_R=Working\_trace\_copy(End\_X)+Hight\_R\*0.25

Variable Three\_fouth\_Height\_Y\_R=Working\_trace\_copy(End\_X)+Hight\_R\*0.75

Variable One\_fouth\_Height\_X\_R, Three\_fouth\_Height\_X\_R

FindLevel /Q/R=(T\_Max[Peak\_pnt],End\_X) Working\_trace\_copy, Three\_fouth\_Height\_Y\_R

Three\_fouth\_Height\_X\_R=V\_LevelX

FindLevel /Q/R=(T\_Max[Peak\_pnt],End\_X) Working\_trace\_copy, One\_fouth\_Height\_Y\_R

One\_fouth\_Height\_X\_R=V\_LevelX

Variable Fall\_DeltaT=(One\_fouth\_Height\_X\_R-Three\_fouth\_Height\_X\_R)\*1000

Variable Fall\_Slope\_coeff, Fall\_Slope\_coeff2=0,Fall\_Chi=1

If(x2pnt(Working\_trace\_copy,End\_X)-x2pnt(Working\_trace\_copy,Three\_fouth\_Height\_X\_R)<5)

return "0,0,0,0"

endif

If(cmpstr(Fit\_Function,"DblExp")==0)

PauseUpdate

Duplicate/O/R=(Three\_fouth\_Height\_X\_R,End\_X) Working\_trace\_copy, Fall\_phase

SetScale/I x 0,(End\_X-Three\_fouth\_Height\_X\_R),"s", Fall\_phase

CurveFit/Q/N dblexp Fall\_phase /D=Fall\_phase

Fall\_Chi=V\_chisq

Fall\_Slope\_coeff=min((1/(W\_coef[4]/1000)),(1/(W\_coef[2]/1000)))

Fall\_Slope\_coeff2=max((1/(W\_coef[4]/1000)),(1/(W\_coef[2]/1000)))

SetScale/I x Three\_fouth\_Height\_X\_R,End\_X,"s", Fall\_phase

else

If(cmpstr(Fit\_Function,"Exp")==0)

PauseUpdate

Duplicate/O/R=(Three\_fouth\_Height\_X\_R,End\_X) Working\_trace\_copy, Fall\_phase

SetScale/I x 0,(End\_X-Three\_fouth\_Height\_X\_R),"s", Fall\_phase

CurveFit/Q/N exp Fall\_phase /D=Fall\_phase

Fall\_Chi=V\_chisq

SetScale/I x Three\_fouth\_Height\_X\_R,End\_X,"s", Fall\_phase

Fall\_Slope\_coeff=1/(W\_coef[2]/1000)

else

PauseUpdate

Duplicate/O/R=(Three\_fouth\_Height\_X\_R,One\_fouth\_Height\_X\_R) Working\_trace\_copy, Fall\_phase

CurveFit/Q/N line Fall\_phase /D=Fall\_phase

Fall\_Slope\_coeff=-W\_coef[1]/1000

endif

endif

String Formated\_String, Slope\_coeffs

sprintf Formated\_String, "%.12f", Fall\_DeltaT

Slope\_coeffs=Formated\_String+";"

sprintf Formated\_String, "%.12f", Fall\_Slope\_coeff

Slope\_coeffs=Slope\_coeffs+Formated\_String+";"

sprintf Formated\_String, "%.12f", Fall\_Slope\_coeff2

Slope\_coeffs=Slope\_coeffs+Formated\_String+";"

sprintf Formated\_String, "%.18f", Fall\_Chi

Slope\_coeffs=Slope\_coeffs+Formated\_String+";"

return Slope\_coeffs

End

Function Calc\_Peak\_Q(Peak\_pnt,Start\_X,End\_X)

Variable Peak\_pnt

Variable Start\_X

Variable End\_X

SetDataFolder $"root:Quanta"

Wave Peak\_Q=Peak\_Q

Wave Peak\_Molec=Peak\_Molec

Wave Working\_trace\_copy=Working\_trace\_copy

Variable Area\_total=area(Working\_trace\_copy,Start\_X,End\_X)

Variable Area\_bkg=(End\_X-Start\_X)\*(Working\_trace\_copy(Start\_X)+Working\_trace\_copy(End\_X))/2

Peak\_Q[Peak\_pnt]=Area\_total-Area\_bkg

Peak\_Molec[Peak\_pnt]=(Peak\_Q[Peak\_pnt])\*3.121\*10^6

End

Function Calc\_Foot\_Parameters(Peak\_pnt) : ButtonControl

Variable Peak\_pnt

SetDataFolder $"root:Quanta"

Wave T\_Max=T\_Max

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Foot\_Q=Foot\_Q

Wave Foot\_Molec=Foot\_Molec

Wave Foot\_W=Foot\_W

Wave Foot\_end=Foot\_end

Wave Rise\_Lowpnt\_X=Rise\_Lowpnt\_X

Wave Foot\_I= Foot\_I

Wave Peak\_Split1=Peak\_Split1

Wave Rise\_time=Rise\_time

Wave Working\_trace\_copy=Working\_trace\_copy

NVAR Bkg\_noise\_I=Bkg\_noise\_I

NVAR Detection\_Foot\_Mult=Detection\_Foot\_Mult

NVAR SSFoot\_Do=SSFoot\_Do

NVAR Native\_Foot\_Del=Native\_Foot\_Del

NVAR Foot\_Min\_W=Foot\_Min\_W

NVAR Rise\_Hi\_Prc=Rise\_Hi\_Prc

NVAR Rise\_Low\_Prc=Rise\_Low\_Prc

Variable Start\_X=T\_Bkg1[Peak\_pnt]

Variable Foot\_End\_X\_low=Foot\_end[Peak\_pnt]

Variable Foot\_End\_X\_hi=Rise\_Lowpnt\_X[Peak\_pnt]

If((Peak\_Split1(Peak\_pnt)!=0)%|(Foot\_End\_X\_low<=Start\_X))

Foot\_Q[Peak\_pnt]=0

Foot\_Molec[Peak\_pnt]=0

Foot\_W[Peak\_pnt]=0

Foot\_I[Peak\_pnt]=0

return 0

endif

// calculate foot duration

FindLevel/Q/R=(Start\_X,T\_Max[Peak\_pnt]) Working\_trace\_copy, (Working\_trace\_copy(Start\_X)+Detection\_Foot\_Mult\*Bkg\_noise\_I)

If((Foot\_End\_X\_low-V\_LevelX)>0)

Foot\_W[Peak\_pnt]=(Foot\_End\_X\_low-Start\_X)\*1000

else

Foot\_W[Peak\_pnt]=0

endif

// delete 'native' PSF

If((Native\_Foot\_Del==1)&(Foot\_W[Peak\_pnt]>Foot\_Min\_W))

Variable Native\_foot\_W=Rise\_time[Peak\_pnt]\*13.2/(Rise\_Hi\_Prc-Rise\_Low\_Prc)

If (Foot\_W[Peak\_pnt]<=Native\_foot\_W)

Foot\_W[Peak\_pnt]=0

endif

endif

// calculate foot current

If((SSFoot\_Do==1)&(Foot\_W[Peak\_pnt]>Foot\_Min\_W))

// find PSF with steady states

Variable SteadyState=Find\_SteadyState(Foot\_End\_X\_low, Start\_X, Foot\_Min\_W/1000)

If(SteadyState)

Foot\_I[Peak\_pnt]=SteadyState-Working\_trace\_copy(Start\_X)

endif

else

Foot\_I[Peak\_pnt]=mean(Working\_trace\_copy, Start\_X, Foot\_End\_X\_low)-Working\_trace\_copy(Start\_X)

endif

// calculate foot charge

If(Foot\_W[Peak\_pnt]>0)

Variable Area\_total=area(Working\_trace\_copy,Start\_X,Foot\_End\_X\_hi)

Variable Ft\_Bkg2\_Y=Y\_offset(Start\_X, T\_Bkg2[Peak\_pnt], Foot\_End\_X\_hi)

Variable Foot\_W\_all=Foot\_End\_X\_hi-Start\_X

Variable Area\_bkg=Foot\_W\_all\*(Working\_trace\_copy(Start\_X)+Ft\_Bkg2\_Y)/2

Area\_total=Area\_total-Area\_bkg

Variable Tr\_Area=(Foot\_End\_X\_hi-Foot\_End\_X\_low)\*(Working\_trace\_copy(Foot\_End\_X\_hi)-Ft\_Bkg2\_Y)/2

Foot\_Q[Peak\_pnt]=Area\_total-Tr\_Area

Foot\_Molec[Peak\_pnt]=(Foot\_Q[Peak\_pnt])\*3.121\*10^6

endif

// check the validity of PSF values

If(Detection\_limits\_Foot(Peak\_pnt))

Foot\_W[Peak\_pnt]=0

Foot\_I[Peak\_pnt]=0

Foot\_Q[Peak\_pnt]=0

Foot\_Molec[Peak\_pnt]=0

endif

End

// All in seconds

Function Find\_SteadyState (Start\_X, End\_X, Itteration)

Variable Start\_X

Variable End\_X

Variable Itteration

Wave Working\_trace\_copy=Working\_trace\_copy

NVAR Bkg\_noise\_I=Bkg\_noise\_I

Variable Delta=Start\_X-End\_X

If(Delta<0) // The search is toward trace's end

Itteration=0-Itteration

endif

If(abs(Delta)<abs(Itteration))

return 0

endif

Variable Step\_back\_1=Start\_X

Variable Steady\_State1=mean(Working\_trace\_copy, Step\_back\_1, Step\_back\_1-Itteration/2)

Variable Steady\_State2=mean(Working\_trace\_copy, Step\_back\_1-Itteration/2, Step\_back\_1-Itteration)

Variable Upper\_lim=Steady\_State1+Bkg\_noise\_I

Variable Lower\_lim=Steady\_State1-Bkg\_noise\_I

Variable Steady\_State

do

If((Steady\_State2<Upper\_lim)&(Steady\_State2>Lower\_lim))

Steady\_State=mean(Working\_trace\_copy, Step\_back\_1, Step\_back\_1-Itteration)

return Steady\_State

break

endif

Step\_back\_1-=Itteration/2

Steady\_State1=mean(Working\_trace\_copy, Step\_back\_1, Step\_back\_1-Itteration/2)

Steady\_State2=mean(Working\_trace\_copy, Step\_back\_1-Itteration/2, Step\_back\_1-Itteration)

Upper\_lim=Steady\_State1+Bkg\_noise\_I

Lower\_lim=Steady\_State1-Bkg\_noise\_I

while(abs(Step\_back\_1-End\_X)>abs(Itteration))

End

Function New\_Foot\_I(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

Wave T\_Bkg1=T\_Bkg1

Wave Foot\_I=Foot\_I

Wave Working\_trace\_copy=Working\_trace\_copy

Variable Peak\_pnt=Peak\_ID-1

Foot\_I[Peak\_pnt]=Working\_trace\_copy(xcsr(A))-Working\_trace\_copy(T\_Bkg1[Peak\_pnt])

Draw\_lines\_All(Peak\_pnt)

End

Function Delete\_Foot(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

Wave T\_Max=T\_Max

Wave T\_Bkg1=T\_Bkg1

Wave Peak\_Half\_H1=Peak\_Half\_H1

Wave Foot\_Q=Foot\_Q

Wave Foot\_Molec=Foot\_Molec

Wave Foot\_W=Foot\_W

Wave Foot\_end=Foot\_end

Wave Rise\_Lowpnt\_X=Rise\_Lowpnt\_X

Wave Rise\_Hipnt\_X=Rise\_Hipnt\_X

Wave Foot\_I=Foot\_I

Wave Rise\_Midpoint=Rise\_Midpoint

Wave Rise\_slope=Rise\_slope

Wave Rise\_time=Rise\_time

Wave Working\_Trace\_Copy=Working\_Trace\_Copy

NVAR Rise\_Hi\_Prc=Rise\_Hi\_Prc

NVAR Rise\_Low\_Prc=Rise\_Low\_Prc

Variable Peak\_pnt=Peak\_ID-1

Rise\_Midpoint[Peak\_pnt]=Peak\_Half\_H1[Peak\_pnt]

Variable Hight\_L=Working\_trace\_copy(T\_Max[Peak\_pnt])-Working\_trace\_copy(T\_Bkg1[Peak\_pnt])

Variable Upper\_Y=Working\_trace\_copy(T\_Bkg1[Peak\_pnt])+Hight\_L\*Rise\_Hi\_Prc/100

Variable Lower\_Y=Working\_trace\_copy(T\_Bkg1[Peak\_pnt])+Hight\_L\*Rise\_Low\_Prc/100

FindLevel/P/Q/R=(T\_Max[Peak\_pnt],T\_Bkg1[Peak\_pnt]) Working\_trace\_copy, Upper\_Y

Rise\_Hipnt\_X[Peak\_pnt]=pnt2x(Working\_trace\_copy, V\_LevelX )

FindLevel/P/Q/R=(T\_Max[Peak\_pnt],T\_Bkg1[Peak\_pnt]) Working\_trace\_copy, Lower\_Y

Rise\_Lowpnt\_X[Peak\_pnt]=pnt2x(Working\_trace\_copy, V\_LevelX )

Variable Rise\_B\_coeff=Fit\_Rise(Rise\_Lowpnt\_X[Peak\_pnt],Rise\_Hipnt\_X[Peak\_pnt])

Rise\_slope[Peak\_pnt]=Rise\_B\_coeff/1000

Foot\_End[Peak\_pnt]=Extrapolate\_to\_Baseline(Peak\_pnt)

Variable Rise\_Segment=(Rise\_Hi\_Prc-Rise\_Low\_Prc)/100

Rise\_time[Peak\_pnt]=(T\_max[Peak\_pnt]-Foot\_end[Peak\_pnt])\*1000\*Rise\_Segment

Foot\_Q[Peak\_pnt]=0

Foot\_W[Peak\_pnt]=0

Foot\_Molec[Peak\_pnt]=0

Foot\_I[Peak\_pnt]=0

Draw\_lines\_All(Peak\_pnt)

End

Function New\_Baseline(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

NVAR Total\_peaks\_number=Total\_peaks\_number

Wave T\_Max=T\_Max

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Rise\_time=Rise\_time

Wave Rise\_Midpoint=Rise\_Midpoint

Wave Rise\_Lowpnt\_X=Rise\_Lowpnt\_X

Wave Rise\_Hipnt\_X=Rise\_Hipnt\_X

Wave Rise\_slope=Rise\_slope

Wave Foot\_end=Foot\_end

Wave/T Fall\_fit=Fall\_fit

Wave Working\_Trace\_Copy=Working\_Trace\_Copy

Variable Peak\_pnt=Peak\_ID-1

If(Total\_peaks\_number==0)

abort

endif

Variable Start\_X=min(xcsr(A),xcsr(B))

Variable End\_X=max(xcsr(A),xcsr(B))

If((T\_Bkg1[Peak\_pnt]>T\_Bkg2[Peak\_pnt-1])%|(Peak\_pnt==0))

FindLevel /Q/R=(T\_Max[Peak\_pnt],Start\_X ) Working\_Trace\_Copy, Working\_Trace\_Copy(Start\_X)

If(V\_flag==0)

Start\_X=V\_LevelX

endif

endif

If ((Start\_X>=T\_Max[Peak\_pnt])%|(End\_X<=T\_Max[Peak\_pnt]))

abort "Spike Maximum has to be between the coursors!"

endif

If ((Peak\_Split1[Peak\_pnt]!=0)%|(Peak\_Split2[Peak\_pnt]!=0))

T\_Bkg1[Peak\_pnt]=Start\_X

T\_Bkg2[Peak\_pnt]=End\_X

Calc\_split\_peak\_param(Peak\_pnt)

else

Fall\_fit[Peak\_pnt]=""

Rise\_Lowpnt\_X[Peak\_pnt]=0

Rise\_Hipnt\_X[Peak\_pnt]=0

Foot\_end[Peak\_pnt]=0

Rise\_slope[Peak\_pnt]=0

Rise\_Midpoint[Peak\_pnt]=0

Calc\_Peak\_Parameters(peak\_pnt,Start\_X,End\_X)

endif

Draw\_lines\_All(Peak\_pnt)

End

Function New\_T\_Max(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

NVAR Total\_peaks\_number=Total\_peaks\_number

Variable Peak\_pnt=Peak\_ID-1

Wave T\_Max=T\_Max

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Rise\_Midpoint=Rise\_Midpoint

Wave Rise\_Lowpnt\_X=Rise\_Lowpnt\_X

Wave Rise\_Hipnt\_X=Rise\_Hipnt\_X

Wave Rise\_slope=Rise\_slope

Wave Foot\_End=Foot\_End

Wave/T Fall\_fit=Fall\_fit

If(Total\_peaks\_number==0)

abort

endif

If ((xcsr(A)<=T\_Bkg1[Peak\_pnt])%|(xcsr(A)>=T\_Bkg2[Peak\_pnt]))

abort "C'mon, the Maximum has to be between the Start and the End on a spike!"

endif

T\_Max[Peak\_pnt]=xcsr(A)

Fall\_fit[Peak\_pnt]=""

Rise\_Lowpnt\_X[Peak\_pnt]=0

Rise\_Hipnt\_X[Peak\_pnt]=0

Rise\_slope[Peak\_pnt]=0

Rise\_Midpoint[Peak\_pnt]=0

Foot\_end[Peak\_pnt]=0

Calc\_Peak\_Parameters(Peak\_pnt,T\_Bkg1[Peak\_pnt],T\_Bkg2[Peak\_pnt])

Draw\_lines\_All(Peak\_pnt)

End

Function Delete\_Peak(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

NVAR Total\_peaks\_number=Total\_peaks\_number

Variable Peak\_pnt=Peak\_ID-1

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Peak\_Num=Peak\_Num

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave T\_Max=T\_Max

Wave/T Fall\_fit\_Extrap=Fall\_fit\_Extrap

If(Total\_peaks\_number==0)

abort

endif

Wavestats/Q Peak\_Num

If( V\_npnts==1)

De\_novo()

else

Variable Over=0

String First\_and\_Last=Check\_for\_Separated\_peaks(Peak\_pnt)

Variable First\_separated\_pnt=str2num(StringFromList(0,First\_and\_Last))

Variable Last\_separated\_pnt=str2num(StringFromList(1,First\_and\_Last))

If(First\_separated\_pnt!=Last\_separated\_pnt)

Variable i=First\_separated\_pnt

Do

Fall\_fit\_Extrap[i]=""

i+=1

while(i<=Last\_separated\_pnt)

Over=1

endif

String TagName="Max"+num2str(Total\_peaks\_number-1)

Tag/K/N=$TagName

If (Peak\_Split1[Peak\_pnt])

Peak\_Split2[Peak\_pnt-1]=Peak\_Split2[Peak\_pnt]

Over=1

endif

If ((Peak\_Split1[Peak\_pnt]==0)&(Peak\_Split2[Peak\_pnt]!=0))

Peak\_Split1[Peak\_pnt+1]=0

Over=1

endif

Change\_waves("Delete",Peak\_pnt)

Wavestats/Q Peak\_Num

Total\_peaks\_number=V\_npnts

If(Over==1)

Calc\_Peak\_Parameters(Peak\_pnt,T\_Bkg1[Peak\_pnt],T\_Bkg2[Peak\_pnt])

Calc\_Peak\_Parameters(Peak\_pnt-1,T\_Bkg1[Peak\_pnt-1],T\_Bkg2[Peak\_pnt-1])

endif

endif

If (Peak\_ID>=Total\_peaks\_number)

Peak\_ID=Total\_peaks\_number

else

Peak\_Num[Peak\_pnt,]-=1

endif

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Main\_window

if (CheckName("Zoom\_Win", 6)!=0)

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Zoom\_Win

endif

if(Total\_peaks\_number)

Draw\_lines\_All(Peak\_ID-1)

endif

End

Function Delete\_All\_Peaks(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

NVAR Total\_peaks\_number=Total\_peaks\_number

If(Total\_peaks\_number==0)

abort

endif

DoAlert 1, "Do you really want to kill them all!"

If (V\_Flag==1)

De\_novo()

else

abort

endif

End

///\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_\_\_\_\_\_\_Displaying the data\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Function Peak\_surf(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

NVAR Total\_peaks\_number=Total\_peaks\_number

If(Total\_peaks\_number>0)

If(cmpstr(ctrlName, "Next")==0)

Peak\_ID+=1

If (Peak\_ID>Total\_peaks\_number)

Peak\_ID=Total\_peaks\_number

endif

else

Peak\_ID-=1

If (Peak\_ID<1)

Peak\_ID=1

endif

endif

else

abort

endif

Peak\_locator(Peak\_ID-1)

End

Function Peak\_locator\_ID(ctrlName,varNum,varStr,varName) : SetVariableControl

String ctrlName

Variable varNum

String varStr

String varName

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

Peak\_locator(Peak\_ID-1)

End

Function Peak\_locator(Peak\_pnt)

Variable Peak\_pnt

SetDataFolder $"root:Quanta"

SVAR Zoom\_On=Zoom\_On

Wave Peak\_Half\_H1=Peak\_Half\_H1

Wave T\_Max=T\_Max

Wave Peak\_Imax=Peak\_Imax

Wave Peak\_Half\_H2=Peak\_Half\_H2

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Working\_trace\_copy=Working\_trace\_copy

Variable Max\_location=T\_Max[Peak\_pnt]

NVAR Total\_peaks\_number=Total\_peaks\_number

If(Total\_peaks\_number==0)

abort

endif

Variable Bkg1\_location=T\_Bkg1[Peak\_pnt]

Variable Tail\_location=T\_Bkg2[Peak\_pnt]

If ((T\_Max[Peak\_pnt])<(T\_Bkg2[Peak\_pnt-1]))

Bkg1\_location=T\_Bkg1[Peak\_pnt-1]

endif

If ((T\_Max[Peak\_pnt])>(T\_Bkg1[Peak\_pnt+1]))

Tail\_location=T\_Bkg2[Peak\_pnt+1]

endif

Variable Bottom\_Y=min(Working\_trace\_copy(Bkg1\_location),Working\_trace\_copy(Tail\_location))

wavestats/Q/R=(Bkg1\_location,Tail\_location) Working\_trace\_copy

Variable Highest\_point\_X=V\_maxloc

Variable Highest\_point\_Y=Working\_trace\_copy(Highest\_point\_X)-Bottom\_Y

If(cmpstr(Zoom\_On,"On")==0)

SetAxis left (Bottom\_Y-0.2\*Highest\_point\_Y),(Working\_trace\_copy(Highest\_point\_X)+0.2\*Highest\_point\_Y)

SetAxis bottom (Bkg1\_location-1\*(Tail\_location-Highest\_point\_X)),(Tail\_location+1\*(Tail\_location-Highest\_point\_X))

endif

Cursor/W=Main\_window A Working\_trace\_copy T\_Bkg1[Peak\_pnt]

Cursor/W=Main\_window B Working\_trace\_copy T\_Bkg2[Peak\_pnt]

Draw\_lines\_All(Peak\_pnt)

Slider\_Reset("qqq")

End

Function Generate\_annotation(Current\_Peak)

Variable Current\_Peak

SetDataFolder $"root:Quanta"

Wave T\_Max=T\_Max

Wave Peak\_Base=Peak\_Base

Wave Peak\_t05=Peak\_t05

Wave Peak\_Imax=Peak\_Imax

Wave Peak\_Q=Peak\_Q

Wave Peak\_Molec=Peak\_Molec

Wave Rise\_time=Rise\_time

Wave Rise\_slope=Rise\_slope

Wave/T Fall\_fit=Fall\_fit

Wave Fall\_time=Fall\_time

Wave Fall\_slope=Fall\_slope

Wave Fall\_slope2=Fall\_slope2

Wave Foot\_I=Foot\_I

Wave Foot\_W=Foot\_W

Wave Foot\_Q=Foot\_Q

Wave Foot\_Molec=Foot\_Molec

NVAR Rise\_Low\_Prc=Rise\_Low\_Prc

NVAR Rise\_Hi\_Prc=Rise\_Hi\_Prc

NVAR Total\_peaks\_number=Total\_peaks\_number

SVAR Values\_to\_show=Values\_to\_show

String Peak\_Parameters\_list="",Peak\_Parameters\_One

String One\_wave

If(Total\_peaks\_number==0)

return 0

endif

Variable i=1

do

One\_wave=StringFromList(i, Values\_to\_show ,",")

if(strlen(One\_wave) == 0 )

break

endif

If(cmpstr(One\_wave,"T\_Max")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\K(0,0,0) Time= %.2W1Ps\K(0,0,52224)", T\_Max[Current\_Peak]

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Peak\_Base")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rBase= %.1W1Ps", Peak\_Base[Current\_Peak]/1000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Peak\_t05")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rt1/2= %.1W1Ps", Peak\_t05[Current\_Peak]/1000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Peak\_Imax")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rImax= %.1W1PA", Peak\_Imax[Current\_Peak]/1000000000000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Peak\_Q")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rQ= %.1W1PC", Peak\_Q[Current\_Peak]/1000000000000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Peak\_Molec")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rQ= %.3g Molec", Peak\_Molec[Current\_Peak]

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Rise\_time")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rrise ("+num2str(Rise\_Low\_Prc)+"-"+num2str(Rise\_Hi\_Prc)+")= %.1W1Ps", Rise\_time[Current\_Peak]/1000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Rise\_slope")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rrise= %.1W1PA/ms", Rise\_slope[Current\_Peak]/1000000000000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Fall\_time")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rfall (75-25)= %.1W1Ps", Fall\_time[Current\_Peak]/1000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Fall\_slope")==0)

SVAR Fit\_method=Fit\_method

Peak\_Parameters\_list+="\K(0,0,0)\r Fit: "+Fall\_fit[Current\_Peak]+"\K(0,0,52224)"

If (cmpstr(Fit\_method,"Line")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rfall= %.1W1PA/ms", Fall\_slope[Current\_Peak]/1000000000000

endif

If (cmpstr(Fit\_method,"Exp")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rfall (\F'Symbol't\F'Arial'1)= %.1W1Ps", Fall\_slope[Current\_Peak]/1000

endif

If (cmpstr(Fit\_method,"DblExp")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rfall (\F'Symbol't\F'Arial'1)= %.1W1Ps \rfall (\F'Symbol't\F'Arial'2)= %.1W1Ps", Fall\_slope[Current\_Peak]/1000, Fall\_slope2[Current\_Peak]/1000

endif

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Foot\_I")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rFoot H= %.1W1PA", Foot\_I[Current\_Peak]/1000000000000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Foot\_W")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rFoot W= %.1W1Ps", Foot\_W[Current\_Peak]/1000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Foot\_Q")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rFoot Q= %.1W1PC", Foot\_Q[Current\_Peak]/1000000000000

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

If(cmpstr(One\_wave,"Foot\_Molec")==0)

sprintf Peak\_Parameters\_One, "\Z09\F'Arial'\rFoot Q= %.3g Molec", Foot\_Molec[Current\_Peak]

Peak\_Parameters\_list+=Peak\_Parameters\_One

endif

i+=1

while(1)

TextBox/W=Main\_window/A=RT/C/N=Peak\_data/F=1/G=(0,0,52224) Peak\_Parameters\_list

End

Function Draw\_lines\_All(Current\_Peak)

Variable Current\_Peak

SetDataFolder $"root:Quanta"

SVAR Zoom\_On=Zoom\_On

NVAR Total\_peaks\_number=Total\_peaks\_number

Wave Working\_trace\_copy=Working\_trace\_copy

Wave T\_Max=T\_Max

Wave Peak\_Half\_H1=Peak\_Half\_H1

Wave Peak\_Half\_H2=Peak\_Half\_H2

Wave Peak\_Imax=Peak\_Imax

Wave T\_Bkg1=T\_Bkg1

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Peak\_t05=Peak\_t05

Wave T\_Bkg2=T\_Bkg2

Wave Rise\_Midpoint=Rise\_Midpoint

Wave Foot\_Q=Foot\_Q

Wave Foot\_W=Foot\_W

Wave Foot\_end=Foot\_end

Wave Rise\_Lowpnt\_X=Rise\_Lowpnt\_X

Wave Rise\_Hipnt\_X=Rise\_Hipnt\_X

DoWindow/F Main\_Window

PauseUpdate; Silent 1

GetAxis /Q left

Variable Mark\_Height=(V\_max-V\_min)\*0.04

SetDrawLayer /K UserFront

Delete\_Tags()

Variable Peak\_pnt=0

String TagName

NVAR Show\_Legend=Show\_Legend

if(Show\_Legend==1)

Generate\_annotation(Current\_Peak)

endif

Do

Variable Max\_X=T\_Max[Peak\_pnt]

Variable Start\_X=Peak\_Half\_H1[Peak\_pnt]

Variable End\_X=Peak\_Half\_H2[Peak\_pnt]

Variable Bkg1\_location=T\_Bkg1[Peak\_pnt]

Variable Tail\_location=T\_Bkg2[Peak\_pnt]

Variable Bottom\_Y=min(Working\_trace\_copy(Bkg1\_location),Working\_trace\_copy(Tail\_location))

TagName="Max"+num2str(Peak\_pnt)

Tag/C/N=$TagName Working\_trace\_copy, (Max\_X), num2str(Peak\_pnt+1)

Tag/C/N=$TagName /F=0 /X=0.00/Y=8

If(Peak\_pnt==Current\_Peak)

Tag/C/N=$TagName/G=(0,15872,65280)

else

Tag/C/N=$TagName /G=(0,0,0)/I=1

endif

String First\_and\_Last=Check\_for\_Separated\_peaks(Peak\_pnt)

Variable First\_separated\_pnt=str2num(StringFromList(0,First\_and\_Last))

Variable Last\_separated\_pnt=str2num(StringFromList(1,First\_and\_Last))

String Extrap\_Wave="Extrap\_"+num2str(Peak\_pnt+1)

String Extrap\_Trace=TraceNameList("Main\_window", ";", 1 )

If(First\_separated\_pnt!=Last\_separated\_pnt)

If(strsearch(Extrap\_Trace,Extrap\_Wave,0)==-1)

If(WaveExists($Extrap\_Wave))

AppendToGraph/W=Main\_window $Extrap\_Wave

ModifyGraph lstyle($Extrap\_Wave)=2,lsize($Extrap\_Wave)=0.5,rgb($Extrap\_Wave)=(0,15872,65280)

endif

endif

If(Peak\_pnt==First\_separated\_pnt)

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,15872,65280), dash= 2

DrawLine Bkg1\_location,Working\_trace\_copy(Bkg1\_location),Tail\_location,Working\_trace\_copy(Tail\_location)

endif

else

If(WaveExists($Extrap\_Wave))

if (CheckName("Zoom\_Win", 6)!=0)

RemoveFromGraph/Z/W=Zoom\_Win $Extrap\_Wave

endif

RemoveFromGraph/Z/W=Main\_window $Extrap\_Wave

KillWaves/Z $Extrap\_Wave

endif

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,15872,65280), dash= 2

DrawLine Bkg1\_location,Working\_trace\_copy(Bkg1\_location),Tail\_location,Working\_trace\_copy(Tail\_location)

endif

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,15872,65280), dash= 2

DrawLine Start\_X,(Working\_trace\_copy(Max\_X)-Peak\_Imax[Peak\_pnt]/2),End\_X,(Working\_trace\_copy(Max\_X)-Peak\_Imax[Peak\_pnt]/2)

if(cmpstr(Zoom\_On,"On")==0 )

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (34816,34816,34816), arrow= 1

DrawLine Bkg1\_location,(Working\_trace\_copy(Bkg1\_location)-Mark\_Height),Bkg1\_location,(Working\_trace\_copy(Bkg1\_location))

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (34816,34816,34816), arrow= 1

DrawLine Tail\_location,(Working\_trace\_copy(Tail\_location)-Mark\_Height),Tail\_location,(Working\_trace\_copy(Tail\_location))

endif

If (Foot\_Q[Peak\_pnt]!=0)

Variable Low\_Rise\_Pnt\_X=Rise\_Lowpnt\_X[Peak\_pnt]

Variable Low\_Rise\_Pnt\_Y=Working\_trace\_copy(Rise\_Lowpnt\_X[Peak\_pnt])

Variable Offset\_Foot=Y\_offset(T\_Bkg1[Peak\_pnt], T\_Bkg2[Peak\_pnt], Foot\_end[Peak\_pnt])

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,39168,0), dash= 2

DrawLine Low\_Rise\_Pnt\_X,Low\_Rise\_Pnt\_Y,Foot\_end[Peak\_pnt],Offset\_Foot

endif

If (Peak\_Split1[Peak\_pnt]!=0)

Variable Offset\_Split1=Y\_offset(T\_Bkg1[Peak\_pnt], T\_Bkg2[Peak\_pnt], Peak\_Split1[Peak\_pnt])

SetDrawEnv xcoord=bottom, ycoord=left, linefgc=(24576,24576,65280), dash=0

DrawLine Peak\_Split1[Peak\_pnt],Working\_trace\_copy(Peak\_Split1[Peak\_pnt]),Peak\_Split1[Peak\_pnt],Offset\_Split1

endif

If (Peak\_Split2[Peak\_pnt]!=0)

Variable Offset\_Split2=Y\_offset(T\_Bkg1[Peak\_pnt], T\_Bkg2[Peak\_pnt], Peak\_Split2[Peak\_pnt])

SetDrawEnv xcoord=bottom, ycoord=left, linefgc=(24576,24576,65280), dash=0

DrawLine Peak\_Split2[Peak\_pnt],Working\_trace\_copy(Peak\_Split2[Peak\_pnt]),Peak\_Split2[Peak\_pnt],Offset\_Split2

endif

Peak\_pnt+=1

while(Peak\_pnt<Total\_peaks\_number)

if((cmpstr(Zoom\_On,"On")==0)%|(CheckName("Zoom\_Win", 6)!=0))

Variable Slope\_coeffs=Fit\_Rise(Rise\_Lowpnt\_X[Current\_Peak],Rise\_Hipnt\_X[Current\_Peak])

Wave/T Fall\_fit=Fall\_fit

String Fit\_Function=Fall\_fit[Current\_Peak]

Fit\_Function=Fit\_Fall\_Decay(Current\_Peak,Fit\_Function)

endif

if(cmpstr(Zoom\_On,"On")==0 )

If (Foot\_W[Current\_Peak])

Tag/C/N=FootLoc Working\_trace\_copy, (Foot\_end[Current\_Peak]), "Foot"

Tag/C/N=FootLoc/F=0 /X=-2.4/Y=11.2

endif

endif

if (CheckName("Zoom\_Win", 6)!=0)

Draw\_lines\_zoom\_window(Current\_Peak)

endif

end

Function Draw\_lines\_zoom\_window(Peak\_pnt)

Variable Peak\_pnt

SetDataFolder $"root:Quanta"

SVAR Zoom\_On=Zoom\_On

Wave Working\_trace\_copy=Working\_trace\_copy

Wave Rise\_phase=Rise\_phase

Wave Fall\_phase=Fall\_phase

Wave T\_Max=T\_Max

Wave Peak\_Half\_H1=Peak\_Half\_H1

Wave Peak\_Half\_H2=Peak\_Half\_H2

Wave Peak\_Imax=Peak\_Imax

Wave T\_Bkg1=T\_Bkg1

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Peak\_t05=Peak\_t05

Wave T\_Bkg2=T\_Bkg2

Wave Rise\_Midpoint=Rise\_Midpoint

Wave Foot\_Q=Foot\_Q

Wave Foot\_W=Foot\_W

Wave Foot\_end=Foot\_end

Wave Rise\_Lowpnt\_X=Rise\_Lowpnt\_X

Wave Rise\_Hipnt\_X=Rise\_Hipnt\_X

Wave Foot\_I=Foot\_I

Variable Max\_X=T\_Max[Peak\_pnt]

Variable Start\_X=Peak\_Half\_H1[Peak\_pnt]

Variable End\_X=Peak\_Half\_H2[Peak\_pnt]

Variable Bkg1\_location=T\_Bkg1[Peak\_pnt]

Variable Tail\_location=T\_Bkg2[Peak\_pnt]

Variable Bottom\_Y=min(Working\_trace\_copy(Bkg1\_location),Working\_trace\_copy(Tail\_location))

PauseUpdate; Silent 1

Dowindow/F Zoom\_Win

GroupBox Separator3,size={189,25},disable=1

Button FootDelete disable=1

Button FootNew\_H disable=1

Button New\_Rise disable=1

PopupMenu Fall\_fit\_change disable=1

SetVariable Extrap\_Tau\_Set disable=1

PopupMenu Fall\_Extrap\_change disable=1

Variable L\_edge=(Bkg1\_location-0.3\*(Tail\_location-Max\_X))

Variable R\_edge=(Tail\_location+0.3\*(Tail\_location-Max\_X))

Duplicate/O/R=(L\_edge,R\_edge) Working\_trace\_copy, Zoomed\_peak

SetAxis left (Bottom\_Y-0.2\*Peak\_Imax[Peak\_pnt]),(Zoomed\_peak(Max\_X)+0.1\*Peak\_Imax[Peak\_pnt])

SetAxis/A Bottom

Cursor A Zoomed\_peak Bkg1\_location;Cursor B Zoomed\_peak Tail\_location

PauseUpdate; Silent 1

SetDrawLayer /K UserFront

Tag/C/N=MaxLoc Zoomed\_peak, (Max\_X), "Max"

Tag/C/N=MaxLoc/F=0 /X=0.00/Y=8

Tag/C/N=StartLoc Zoomed\_peak, (Bkg1\_location), "Start"

Tag/C/N=StartLoc/F=0 /X=0.00/Y=-10.00

Tag/C/N=EndLoc Zoomed\_peak, (Tail\_location), "End"

Tag/C/N=EndLoc/F=0 /X=0.00/Y=-10.00

String All\_Traces=TraceNameList("Zoom\_Win",";", 1)

Variable i=0

Do

String Extrap\_Traces=Stringfromlist(i,All\_Traces)

If(stringmatch(Extrap\_Traces, "Extrap\*")==1)

RemoveFromGraph/Z/W=Zoom\_Win $Extrap\_Traces

endif

i+=1

If(strlen(Extrap\_Traces)==0)

break

endif

while(1)

String First\_and\_Last=Check\_for\_Separated\_peaks(Peak\_pnt)

Variable First\_separated\_pnt=str2num(StringFromList(0,First\_and\_Last))

Variable Last\_separated\_pnt=str2num(StringFromList(1,First\_and\_Last))

i=First\_separated\_pnt

If(First\_separated\_pnt!=Last\_separated\_pnt)

If(Peak\_pnt==First\_separated\_pnt)

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,15872,65280), dash= 2

DrawLine Bkg1\_location,Zoomed\_peak(Bkg1\_location),Tail\_location,Zoomed\_peak(Tail\_location)

endif

Do

String Extrapolation\_name="Extrap\_"+num2str(i+1)

If(WaveExists($Extrapolation\_name))

AppendToGraph/W=Zoom\_Win $Extrapolation\_name

ModifyGraph lstyle($Extrapolation\_name)=2,lsize($Extrapolation\_name)=0.5,rgb($Extrapolation\_name)=(0,15872,65280)

endif

i+=1

while(i<=Last\_separated\_pnt)

SetAxis bottom L\_edge,R\_edge

else

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,15872,65280), dash= 2

DrawLine Bkg1\_location,Zoomed\_peak(Bkg1\_location),Tail\_location,Zoomed\_peak(Tail\_location)

endif

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,15872,65280), dash= 2

DrawLine Start\_X,(Zoomed\_peak(Max\_X)-Peak\_Imax[Peak\_pnt]/2),End\_X,(Zoomed\_peak(Max\_X)-Peak\_Imax[Peak\_pnt]/2)

If (Peak\_Split1[Peak\_pnt]!=0)

Bkg1\_location=Peak\_Split1[Peak\_pnt]

Variable Offset\_Split1=Y\_offset(T\_Bkg1[Peak\_pnt], T\_Bkg2[Peak\_pnt], Peak\_Split1[Peak\_pnt])

SetDrawEnv xcoord=bottom, ycoord=left, linefgc=(24576,24576,65280), dash=0

DrawLine Peak\_Split1[Peak\_pnt],Working\_trace\_copy(Peak\_Split1[Peak\_pnt]),Peak\_Split1[Peak\_pnt],Offset\_Split1

endif

If (Peak\_Split2[Peak\_pnt]!=0)

Tail\_location=Peak\_Split2[Peak\_pnt]

Variable Offset\_Split2=Y\_offset(T\_Bkg1[Peak\_pnt], T\_Bkg2[Peak\_pnt], Peak\_Split2[Peak\_pnt])

SetDrawEnv xcoord=bottom, ycoord=left, linefgc=(24576,24576,65280), dash=0

DrawLine Peak\_Split2[Peak\_pnt],Working\_trace\_copy(Peak\_Split2[Peak\_pnt]),Peak\_Split2[Peak\_pnt],Offset\_Split2

endif

If (Foot\_W[Peak\_pnt])

Variable Low\_Rise\_Pnt\_X=Rise\_Lowpnt\_X[Peak\_pnt]

Variable Low\_Rise\_Pnt\_Y=Zoomed\_peak(Rise\_Lowpnt\_X[Peak\_pnt])

Variable Offset\_Foot=Y\_offset(T\_Bkg1[Peak\_pnt], T\_Bkg2[Peak\_pnt], Foot\_end[Peak\_pnt])

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,39168,0), dash= 2,linethick= 2.00

DrawLine Low\_Rise\_Pnt\_X,Low\_Rise\_Pnt\_Y,Foot\_end[Peak\_pnt],Offset\_Foot

Tag/K/N=FootStart

Tag/C/N=FootEnd Zoomed\_peak, (Foot\_end[Peak\_pnt]), "Foot"

Tag/C/N=FootEnd/F=0 /X=-10/Y=8

ControlInfo /W=Zoom\_win Zoom\_to\_Foot

If(V\_Value==1)

Variable Rise\_25=(Rise\_Midpoint[Peak\_pnt]+Rise\_Lowpnt\_X[Peak\_pnt])/2

SetAxis left (Bottom\_Y-0.1\*Peak\_Imax[Peak\_pnt]),Zoomed\_peak(Rise\_25)

SetAxis bottom Bkg1\_location,Rise\_25

Variable Foot\_Start\_X=Foot\_end[Peak\_pnt]-Foot\_W[Peak\_pnt]/1000

Offset\_Foot=Y\_offset(T\_Bkg1[Peak\_pnt], T\_Bkg2[Peak\_pnt], Foot\_Start\_X)

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,39168,0), dash= 1,linethick= 1.00

DrawLine Foot\_Start\_X,(Zoomed\_peak(Bkg1\_location)+Foot\_I[Peak\_pnt]),Foot\_Start\_X,Offset\_Foot

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,39168,0), dash= 1,linethick= 1.00

DrawLine Foot\_end[Peak\_pnt],Zoomed\_peak(Foot\_end[Peak\_pnt]),Foot\_end[Peak\_pnt],Offset\_Foot

SetDrawEnv xcoord=bottom, ycoord=left, linefgc= (0,39168,0), dash= 2,linethick= 2.00

DrawLine Foot\_Start\_X,(Zoomed\_peak(Bkg1\_location)+Foot\_I[Peak\_pnt]),Foot\_end[Peak\_pnt],(Zoomed\_peak(Bkg1\_location)+Foot\_I[Peak\_pnt])

Tag/C/N=FootStart Zoomed\_peak, (Foot\_Start\_X), "Foot Start"

Tag/C/N=FootStart/F=0 /X=0/Y=15

Tag/C/N=FootEnd Zoomed\_peak, (Foot\_end[Peak\_pnt]), "Foot End"

Tag/C/N=FootEnd/F=0 /X=0/Y=15

Tag/K/N=MaxLoc

Tag/K/N=EndLoc

FindLevel /Q/R=(Bkg1\_location,Max\_X ) Zoomed\_peak, (Zoomed\_peak(Bkg1\_location)+Foot\_I[Peak\_pnt])

Cursor A Zoomed\_peak V\_LevelX;Cursor B Zoomed\_peak Foot\_end[Peak\_pnt]

GroupBox Separator3,size={127,23},disable=0

Button FootDelete disable=0

Button FootNew\_H disable=0

Button New\_Rise disable=1

endif

else

Tag/K/N=FootStart

Tag/K/N=FootEnd

endif

ControlInfo /W=Zoom\_win Zoom\_to\_Rise

If(V\_Value==1)

SetAxis bottom Bkg1\_location,Max\_X

Cursor A Zoomed\_peak Rise\_Lowpnt\_X[Peak\_pnt];Cursor B Zoomed\_peak Rise\_Hipnt\_X[Peak\_pnt]

GroupBox Separator3,size={127,23},disable=0

Button FootDelete disable=0

Button FootNew\_H disable=1

Button New\_Rise disable=0

endif

ControlInfo /W=Zoom\_win Zoom\_to\_Fall

If(V\_Value==1)

Wave Fall\_ChiRatio=Fall\_ChiRatio

Wave/T Fall\_fit=Fall\_fit

Wave/T Fall\_fit\_Extrap

String Fit\_method=Fall\_fit[Peak\_pnt]

NVAR Fall\_Tau\_Extrap=Fall\_Tau\_Extrap

PopupMenu Fall\_fit\_change disable=0

If(cmpstr(Fit\_method,"Line")==0)

PopupMenu Fall\_fit\_change,mode=1,popvalue="Line", win=Zoom\_Win

else

If(cmpstr(Fit\_method,"Exp")==0)

PopupMenu Fall\_fit\_change,mode=2,popvalue="Exp", win=Zoom\_Win

else

PopupMenu Fall\_fit\_change,mode=3,popvalue="DblExp", win=Zoom\_Win

Fit\_method="Chi2 ratio= "+num2str(Fall\_ChiRatio[Peak\_pnt])

TextBox/W=Zoom\_Win/A=RT/C/N=Peak\_data/F=0/G=(0,0,52224) Fit\_method

// DrawRect 1.056,0.128,0.41,-0.0575

endif

endif

String Fit\_Extrap\_method=StringByKey("Fit", Fall\_fit\_Extrap[Peak\_pnt])

If(cmpstr(Fit\_Extrap\_method,"")!=0)

PopupMenu Fall\_Extrap\_change disable=0

if(cmpstr(Fit\_Extrap\_method,"Line")!=0)

PopupMenu Fall\_Extrap\_change,mode=2,popvalue="Exp", win=Zoom\_Win

Fall\_Tau\_Extrap=str2num(StringByKey("Tau", Fall\_fit\_Extrap[Peak\_pnt]))

SetVariable Extrap\_Tau\_Set disable=0

else

PopupMenu Fall\_Extrap\_change,mode=1,popvalue="Line", win=Zoom\_Win

endif

DrawRect 1.05,0.36,0.41,-0.0575

else

DrawRect 1.05,0.128,0.41,-0.0575

endif

SetAxis bottom Max\_X,Tail\_location

else

TextBox/K/W=Zoom\_Win/N=Peak\_data

endif

End

Function Delete\_Tags()

SetDataFolder $"root:Quanta"

NVAR Total\_peaks\_number=Total\_peaks\_number

String TagName

Variable i=0

Do

TagName="Max"+num2str(i)

Tag/W=Main\_window/K/N=$TagName

i+=1

while(i<Total\_peaks\_number+1)

Tag/W=Main\_window/K/N=FootLoc

if (CheckName("Zoom\_Win", 6)!=0)

DoWindow/F Zoom\_Win

SetDrawLayer /K UserFront

Tag/K/N=FootStart

Tag/K/N=FootEnd

Tag/K/N=MaxLoc

Tag/K/N=StartLoc

Tag/K/N=EndLoc

TextBox/K/N=Peak\_data

DoWindow/F Main\_Window

endif

End

Function Y\_offset(Bkg1, Bkg2, Med)

Variable Bkg1, Bkg2, Med

SetDataFolder $"root:Quanta"

Wave Working\_trace\_copy=Working\_trace\_copy

Variable Bkg\_H=(Working\_trace\_copy(Bkg2)-Working\_trace\_copy(Bkg1))

Variable Bkg\_W=(Bkg2-Bkg1)

Variable Offset=(Med-Bkg1)\*Bkg\_H/Bkg\_W+Working\_trace\_copy(Bkg1)

return Offset

End

Function Zoom\_Trace\_In\_Out(theTag) : ButtonControl

String theTag

SetDataFolder $"root:Quanta"

SVAR Zoom\_On=Zoom\_On

NVAR Peak\_ID=Peak\_ID

NVAR X\_min=X\_min

NVAR X\_max=X\_max

NVAR Y\_min=Y\_min

NVAR Y\_max=Y\_max

NVAR Peak\_ID=Peak\_ID

NVAR Total\_peaks\_number=Total\_peaks\_number

If(Total\_peaks\_number==0)

abort

endif

strswitch (theTag)

case "See\_all":

Button See\_all rename=See\_zoomed, title="Zoom Out"

Zoom\_On ="On"

GetAxis /Q bottom

X\_min=V\_min

X\_max=V\_max

GetAxis /Q left

Y\_min=V\_min

Y\_max=V\_max

AppendToGraph Rise\_phase,Fall\_phase

ModifyGraph rgb(Rise\_phase)=(0,0,0),rgb(Fall\_phase)=(0,0,0)

break

case "See\_zoomed":

Button See\_zoomed rename=See\_all, title="Zoom In"

Zoom\_On ="Off"

SetAxis bottom X\_min,X\_max

SetAxis left Y\_min,Y\_max

RemoveFromGraph/Z Rise\_phase,Fall\_phase

break

case "See\_foot":

break

endswitch

DoWindow/F Main\_window

Peak\_locator(Peak\_ID-1)

End

Function Zoom\_Down\_Q(ctrlName) : ButtonControl

String ctrlName

Variable Min\_Max\_Delta

GetAxis /Q left

Min\_Max\_Delta=V\_max-V\_min

SetAxis left (V\_min-Min\_Max\_Delta/4),(V\_max+Min\_Max\_Delta/4)

End

Function Zoom\_Up\_Q(ctrlName) : ButtonControl

String ctrlName

Variable Min\_Max\_Delta

GetAxis /Q left

Min\_Max\_Delta=V\_max-V\_min

SetAxis left (V\_min+Min\_Max\_Delta/5),(V\_max-Min\_Max\_Delta/5)

End

Function Zoom\_In\_Horiz\_Q(ctrlName) : ButtonControl

String ctrlName

Variable Min\_Max\_Delta

GetAxis /Q bottom

Min\_Max\_Delta=V\_max-V\_min

SetAxis bottom (V\_min+Min\_Max\_Delta/6),(V\_max-Min\_Max\_Delta/6)

Slider\_Reset(ctrlName)

End

Function Zoom\_OutHoriz\_Q(ctrlName) : ButtonControl

String ctrlName

Variable Min\_Max\_Delta

GetAxis /Q bottom

Min\_Max\_Delta=V\_max-V\_min

SetAxis bottom (V\_min-Min\_Max\_Delta/5),(V\_max+Min\_Max\_Delta/5)

Slider\_Reset(ctrlName)

End

Function Move\_Up\_Q(ctrlName) : ButtonControl

String ctrlName

Variable Min\_Max\_Delta

GetAxis /Q left

Min\_Max\_Delta=V\_max-V\_min

SetAxis left (V\_min+Min\_Max\_Delta/6),(V\_max+Min\_Max\_Delta/6)

End

Function Move\_Down\_Q(ctrlName) : ButtonControl

String ctrlName

Variable Min\_Max\_Delta

GetAxis /Q left

Min\_Max\_Delta=V\_max-V\_min

SetAxis left (V\_min-Min\_Max\_Delta/6),(V\_max-Min\_Max\_Delta/6)

End

Function Move\_Left\_Q(ctrlName) : ButtonControl

String ctrlName

Variable Min\_Max\_Delta

GetAxis /Q bottom

Min\_Max\_Delta=V\_max-V\_min

SetAxis bottom (V\_min-Min\_Max\_Delta/2),(V\_max-Min\_Max\_Delta/2)

Slider\_Reset(ctrlName)

End

Function Move\_Right\_Q(ctrlName) : ButtonControl

String ctrlName

Variable Min\_Max\_Delta

GetAxis /Q bottom

Min\_Max\_Delta=V\_max-V\_min

SetAxis bottom (V\_min+Min\_Max\_Delta/2),(V\_max+Min\_Max\_Delta/2)

Slider\_Reset(ctrlName)

End

Function Slider\_Reset(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

GetAxis/W=Main\_window /Q bottom

Variable dx= (V\_max+V\_min)/2

NVAR T\_Start=T\_Start

NVAR T\_End=T\_End

Variable Slider\_value=(dx-T\_Start)/(T\_End-T\_Start)

Slider X\_Slider,value= Slider\_value,win=Main\_window

End

Function Slider\_Horiz\_Q(TheTag, Value, event)

String TheTag

Variable Value

Variable event

SetDataFolder $"root:Quanta"

String Traces\_Names=TraceNameList("","",1)

String Trace\_Name=StringFromList(0,Traces\_Names)

Variable First\_X=pnt2x($Trace\_Name,0)

Variable Last\_X=pnt2x($Trace\_Name,numpnts($Trace\_Name))

GetAxis/Q bottom

Variable Med\_X= Value\*(Last\_X-First\_X)+First\_X

Variable dX= (V\_max-V\_min)/2

SetAxis bottom,Med\_X-dX,Med\_X+dX

return 0

End

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_Load, Save, Delete Traces\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Window File\_Q() : Panel

PauseUpdate; Silent 1

NewPanel /W=(4,50,729,76) as "Menu"

SetDrawLayer UserBack

SetDrawEnv fillfgc= (39168,0,0)

DrawRect 2,25,242,0

Button Load\_File,pos={5,3},size={83,20},proc=File\_macro\_Q,title="Open File"

PopupMenu Recent\_files,pos={89,2},size={72,21},proc=Recent\_files\_list\_Q,title="Recent"

PopupMenu Recent\_files,mode=0,value= #"root:Quanta:File\_list"

Button Reuse,pos={170,3},size={70,20},proc=File\_macro\_Q,title="Revert"

SetDrawEnv fillfgc= (0,26112,0)

DrawRect 244,0,550,25

Button Menu\_FiltersScales,pos={247,3},size={80,20},proc=Show\_options\_panel,title="Filters/Scales"

Button Menu\_Cutoffs,pos={328,3},size={73,20},proc=Show\_options\_panel,title="Cutoffs"

Button Menu\_Results,pos={402,3},size={73,20},proc=Show\_options\_panel,title="Results"

Button Menu\_Stats,pos={475,3},size={73,20},proc=Show\_options\_panel,title="Stats"

SetDrawEnv fillpat= 0

DrawRect 552,0,723,25

Button Legend\_On,pos={554,3},size={83,20},proc=Show\_Extras,title="Hide Legend"

Button Zoom\_Off,pos={638,3},size={83,20},proc=Show\_Extras,title="Zoom Win"

EndMacro

Macro Load\_single\_file\_Q(From\_the\_list)

Variable From\_the\_list

Close /A

SetDataFolder $"root:Quanta"

Variable/G Peak\_ID

Variable/G Total\_peaks\_number

Variable/G T\_Start\_orig

Variable/G T\_Delta\_orig

String/G File\_to\_load

String/G Loaded\_file\_path

String/G Zoom\_On

variable RefNumber

If(From\_the\_list==1)

KillWaves/Z Orig\_trace\_copy

LoadWave/Q/H/O File\_to\_load

Loaded\_file\_path=File\_to\_load

else

If(cmpstr(igorInfo(2),"Macintosh")==0)

DoAlert 2, "Is it a Macintosh file You are trying to open?\r Press 'No' if the recording was made on a PC"

If (V\_flag==1)

Open/D/R/T="IGBW" RefNumber

else

If (V\_flag==2)

Open/D/R/T="????" RefNumber

else

abort

endif

endif

else

Open/D/R/T="bwav" RefNumber

endif

if(strlen(S\_filename)==0)

abort

endif

Loaded\_file\_path=S\_fileName

KillWaves/Z Orig\_trace\_copy

LoadWave/Q/H/O Loaded\_file\_path

endif

Dowindow/F Main\_window

De\_novo()

String Orig\_Trace\_name=StringFromList(0,S\_waveNames,";")

Duplicate/O $Orig\_Trace\_name Orig\_trace\_copy

KillWaves/Z $Orig\_Trace\_name

T\_Start\_orig=pnt2x(Orig\_trace\_copy,0)

T\_Delta\_orig=(pnt2x(Orig\_trace\_copy,1)-pnt2x(Orig\_trace\_copy,0))\*1000

String info =WaveInfo(Orig\_trace\_copy, 0)

String X\_scale\_units=StringByKey("XUNITS", info)

If(cmpstr(X\_scale\_units,"ms")==0)

T\_Delta\_orig=T\_Delta\_orig/1000

endif

If(cmpstr(X\_scale\_units,"min")==0)

T\_Delta\_orig=T\_Delta\_orig\*1000\*60

endif

Copy\_Orig\_Wave()

String/G Loaded\_file\_name=S\_filename

If(strlen(Loaded\_file\_name)>40)

Loaded\_file\_name=Long\_Name\_Cut(Loaded\_file\_name)

endif

DoWindow/T Main\_window, Loaded\_file\_name

If (CheckName("See\_zoomed", 15)!=0)

Button See\_zoomed rename=See\_all, title="Zoom In"

Zoom\_On ="Off"

endif

If(Check\_files\_list\_Q(Loaded\_file\_path)==0)

String/G File\_list=File\_list+";"+Loaded\_file\_path

String Too\_much=StringFromList(21,File\_list)

if( strlen(Too\_much) != 0 )

string First\_item=StringFromList(1,File\_list)

File\_list=RemoveFromList(First\_item, File\_list)

endif

if (CheckName("File",9)!=0)

PopupMenu Recent\_files,mode=0,value= root:Quanta:File\_list,win=File

endif

endif

Endmacro

Function/S Long\_Name\_Cut(Name)

String Name

Prompt Name, "New name (less than 40 characters)"

DoPrompt "Window name is too long!", Name

Return Name

End

Function Copy\_Orig\_Wave()

SetDataFolder $"root:Quanta"

NVAR Gain=Gain

NVAR Gain\_Temp=Gain\_Temp

NVAR T\_Start\_orig=T\_Start\_orig

NVAR T\_Delta\_orig=T\_Delta\_orig

NVAR T\_Start=T\_Start

NVAR T\_End=T\_End

NVAR T\_Delta=T\_Delta

NVAR X\_min=X\_min

NVAR X\_max=X\_max

NVAR Y\_min=Y\_min

NVAR Y\_max=Y\_max

NVAR Bkg\_noise\_I=Bkg\_noise\_I

NVAR Bkg\_noise\_dI=Bkg\_noise\_dI

NVAR Bkg\_noise\_Start=Bkg\_noise\_Start

NVAR Bkg\_noise\_End=Bkg\_noise\_End

NVAR Overall\_Filter=Overall\_Filter

SVAR Zoom\_On=Zoom\_On

Wave Orig\_trace\_copy=Orig\_trace\_copy

Overall\_Filter=0

Orig\_trace\_copy\*=Gain

If(numpnts(Orig\_trace\_copy)==0)

abort "The trace you are trying to load has 0 datapoints!"

endif

Duplicate/O Orig\_trace\_copy Working\_trace\_copy

Make/O/N=1 Zoomed\_peak, Fall\_phase,Rise\_phase

If(cmpstr(Zoom\_On,"On")==0)

Button See\_zoomed win=Main\_window, rename=See\_all, title="Zoom In"

Zoom\_On ="Off"

RemoveFromGraph/W=Main\_window/Z Rise\_phase,Fall\_phase

endif

T\_Start=T\_Start\_orig

T\_Delta=T\_Delta\_orig

SetScale/P x T\_Start,T\_Delta/1000,"s", Working\_trace\_copy

SetScale d 0,0,"pA", Working\_trace\_copy

T\_End=pnt2x(Working\_trace\_copy, (numpnts(Working\_trace\_copy)-1))

X\_min=T\_Start

X\_max=T\_End

SetAxis/W=Main\_window/A

DoUpdate

GetAxis/W=Main\_window/Q left

Y\_min=V\_min

Y\_max=V\_max

Slider X\_Slider, win=Main\_window,value=0.5

GroupBox Bkg\_HiLt, win=Main\_window,disable=0

Filter\_Limits()

Bkg\_noise\_I=0

Bkg\_noise\_dI=0

Bkg\_noise\_Start=0

Bkg\_noise\_End=0

End

Function File\_macro\_Q(theTag) : ButtonControl

String theTag

SetDataFolder $"root:Quanta"

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Smoothing\_Factor\_diff1=Smoothing\_Factor\_diff1

NVAR Gain=Gain

NVAR T\_Start\_orig=T\_Start\_orig

NVAR T\_delta\_orig=T\_delta\_orig

if( cmpstr(theTag,"Load\_File")==0 )

execute "load\_single\_file\_Q(0)"

endif

if( cmpstr(theTag,"Reuse")==0 )

Wave Orig\_trace\_copy=Orig\_trace\_copy

Wave Working\_trace\_copy=Working\_trace\_copy

if(Total\_peaks\_number)

DoAlert 1, "All detected spikes will be deleted!!! \rDo you really wanna to kill them all!?"

If (V\_Flag==2)

abort

endif

else

If(numpnts(Working\_trace\_copy)<2)

abort "No trace loaded or the wave is too short."

endif

endif

SetScale/P x T\_Start\_orig,T\_Delta\_orig/1000,"s", Orig\_trace\_copy

Orig\_trace\_copy/=Gain

De\_novo()

Copy\_Orig\_Wave()

endif

End

Function Check\_files\_list\_Q(File\_name)

String File\_name

SetDataFolder $"root:Quanta"

SVAR File\_list=File\_list

String File

Variable i=1

do

File=StringFromList(i,File\_list)

if( strlen(File) == 0 )

return 0

break

else

If (cmpstr(File\_name,File)==0)

return 1

break

endif

endif

i += 1

while (1)

End

Function Recent\_files\_list\_Q(ctrlName,popNum,popStr) : PopupMenuControl

String ctrlName

Variable popNum

String popStr

SetDataFolder $"root:Quanta"

SVAR File\_list=File\_list

String/G File\_to\_load

File\_to\_load=popStr

execute "Load\_single\_file\_Q(1)"

End

Function Save\_zoomed\_trace(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

Wave Working\_trace\_copy=Working\_trace\_copy

If(numpnts(Working\_trace\_copy)<2)

abort "No trace loaded or the wave is too short."

endif

SVAR Loaded\_file\_name=Loaded\_file\_name

String Name\_to\_save=Loaded\_file\_name[0,(strsearch(Loaded\_file\_name, ".", 0)-1)]

If(cmpstr(ctrlName,"Save\_Avg")==0)

NVAR Avg\_Spike\_Weight=Avg\_Spike\_Weight

Duplicate/O Avg\_peak Avg\_peak\_saved

Name\_to\_save+="\_Avg\_N"+num2str(Avg\_Spike\_Weight)

Save/C/I Avg\_peak\_saved as Name\_to\_save

KillWaves/Z Avg\_peak\_saved

return 0

endif

GetAxis /Q bottom

variable Start\_X=V\_min

variable End\_X=V\_max

Duplicate/O/R=(Start\_X,End\_X) Working\_trace\_copy zoomed\_trace

Name\_to\_save+="\_Zoom"

Save/C/I zoomed\_trace as Name\_to\_save

End

Function De\_novo()

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

NVAR Total\_peaks\_number=Total\_peaks\_number

Delete\_Tags()

TextBox/W=Main\_window/N=Peak\_data/K

Change\_waves("Make",0)

Total\_peaks\_number=0

Peak\_ID=0

Dowindow/F Main\_window

RemoveFromGraph/Z/W=Main\_window Rise\_phase,Fall\_phase

Make/O/N=1 Zoomed\_peak, Fall\_phase,Rise\_phase

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Main\_window

SetDrawLayer /K UserFront

if (CheckName("Zoom\_Win", 6)!=0)

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Zoom\_Win

endif

String Extrap\_

String Extrap\_waves=WaveList("Extrap\*",";","")

Variable q=0

Do

Extrap\_=Stringfromlist(q,Extrap\_waves)

If(strlen(Extrap\_)==0)

break

endif

if (CheckName("Zoom\_Win", 6)!=0)

RemoveFromGraph/Z/W=Zoom\_Win $Extrap\_

endif

RemoveFromGraph/Z/W=Main\_window $Extrap\_

KillWaves /Z $Extrap\_

q+=1

while(1)

End

Function Change\_Waves(ToDo,pnt)

String ToDo

Variable pnt

SetDataFolder $"root:Quanta"

String/G All\_waves=" Peak\_Num,T\_Max,Peak\_Base,Peak\_Imax,Peak\_t05,Peak\_Q,Peak\_Molec,Rise\_slope,Fall\_slope,Fall\_slope2,"

All\_waves+="Peak\_Half\_H1,Peak\_Half\_H2, T\_Bkg1,T\_Bkg2,Rise\_Midpoint,Rise\_time,Fall\_time,Interspike\_interval,"

All\_waves+="Peak\_Split2, Peak\_Split1,Foot\_Q,Foot\_Molec,Foot\_W,Foot\_end, Rise\_Lowpnt\_X, Rise\_Hipnt\_X, Foot\_I, Fall\_ChiRatio"

String Exe

if( cmpstr(ToDo,"Make")==0)

Exe="Make/O/N="+num2str(pnt)+All\_waves

execute Exe

Exe="Make/O/T/N="+num2str(pnt)+"Fall\_fit, Fall\_Fit\_Extrap"

execute Exe

endif

if( cmpstr(ToDo,"Redimension")==0 )

Exe="Redimension/N="+num2str(pnt)+All\_waves

execute Exe

Exe="Redimension/N="+num2str(pnt)+"Fall\_fit, Fall\_Fit\_Extrap"

execute Exe

endif

String New\_Name="", Extrap\_waves=""

Variable Name\_Ln, Extrap\_Num,q

if( cmpstr(ToDo,"Delete")==0 )

String Extrap\_Exists="Extrap\_"+num2str(pnt+1)

If(exists(Extrap\_Exists)==1)

if (CheckName("Zoom\_Win", 6)!=0)

RemoveFromGraph/Z/W=Zoom\_Win $Extrap\_Exists

endif

RemoveFromGraph/Z/W=Main\_window $Extrap\_Exists

KillWaves/Z $Extrap\_Exists

endif

Extrap\_Exists="Extrap\_"+num2str(pnt)

If(exists(Extrap\_Exists)==1)

if (CheckName("Zoom\_Win", 6)!=0)

RemoveFromGraph/Z/W=Zoom\_Win $Extrap\_Exists

endif

RemoveFromGraph/Z/W=Main\_window $Extrap\_Exists

KillWaves/Z $Extrap\_Exists

endif

Exe="DeletePoints "+num2str(pnt)+",1, "+All\_waves

execute Exe

Exe="DeletePoints "+num2str(pnt)+",1, Fall\_fit, Fall\_Fit\_Extrap"

execute Exe

Extrap\_waves=WaveList("Extrap\*",";","")

q=0

Do

Extrap\_Exists=Stringfromlist(q,Extrap\_waves)

Name\_Ln=strlen(Extrap\_Exists)

If(Name\_Ln==0)

break

endif

Extrap\_Num=str2num(Extrap\_Exists[7,Name\_Ln])

If(Extrap\_Num>=pnt+1)

New\_Name="Extrap\_"+num2str(Extrap\_Num-1)

Rename $Extrap\_Exists, $New\_Name

endif

q+=1

while(1)

endif

if(cmpstr(ToDo,"Insert")==0 )

Exe="InsertPoints "+num2str(pnt)+",1, "+All\_waves

execute Exe

Exe="InsertPoints "+num2str(pnt)+",1, Fall\_fit, Fall\_Fit\_Extrap"

execute Exe

Extrap\_waves=WaveList("Extrap\*",";","")

q=ItemsInList(Extrap\_waves)-1

Do

Extrap\_Exists=Stringfromlist(q,Extrap\_waves)

Name\_Ln=strlen(Extrap\_Exists)

If(Name\_Ln==0)

break

endif

Extrap\_Num=str2num(Extrap\_Exists[7,Name\_Ln])

If(Extrap\_Num>=pnt+1)

New\_Name="Extrap\_"+num2str(Extrap\_Num+1)

Rename $Extrap\_Exists, $New\_Name

endif

q-=1

while(1)

endif

End

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_\_\_\_\_\_Averaged Spike \_\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Function Stats\_PopMenu(theTag,popNum,popStr) : PopupMenuControl

String theTag

Variable popNum

String popStr

SetDataFolder $"root:Quanta"

SVAR Norm\_point=Norm\_point

SVAR Population\_Center=Population\_Center

strswitch(theTag)

case "Stats\_Population":

Population\_Center=popStr

break

case "Stats\_AvePeak":

Norm\_point=popStr

break

endswitch

End

Function Average\_peaks(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

NVAR Total\_peaks\_number=Total\_peaks\_number

Wave Rise\_Midpoint=Rise\_Midpoint

Wave T\_Max=T\_Max

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Peak\_Split1=Peak\_Split1

Wave Peak\_Split2=Peak\_Split2

Wave Fall\_slope=Fall\_slope

Wave Fall\_slope2=Fall\_slope2

Wave Working\_Trace\_Copy=Working\_Trace\_Copy

NVAR Bkg\_noise\_I=Bkg\_noise\_I

NVAR T\_Delta=T\_Delta

NVAR Overall\_Filter=Overall\_Filter

Variable/G Avg\_Spike\_Weight

if (CheckName("Avg\_peak\_graph", 6)==0)

execute "Avg\_peak\_graph()"

else

Dowindow/F Avg\_peak\_graph

MoveWindow 1, 1, 1, 1

endif

if (CheckName("Avg\_peak\_pnts", 6)==0)

PauseUpdate; Silent 1

Edit/K=1/W=(420,259.25,580.5,482.75) Avg\_peak.xy as "Averaged wave"

String command="ModifyTable width(Point)=0,size(Avg\_peak.xy)=9,width(Avg\_peak.xy)=68"

execute command

DoWindow/C Avg\_peak\_pnts

AutoPositionWindow/E/M=1/R=Avg\_peak\_graph Avg\_peak\_pnts

else

Dowindow/F Avg\_peak\_pnts

MoveWindow 1, 1, 1, 1

endif

SetWindow Avg\_peak\_graph, hook=$"Killer\_of\_Hookers"

SetWindow Avg\_peak\_pnts, hook=Killer\_of\_Hookers

if(Total\_peaks\_number==0)

abort

endif

If(numpnts(Avg\_peak)!=0)

DoAlert 1, "Discard existing averaged spike?"

if(V\_flag==1)

Make/O/N=0 Avg\_peak

else

abort

endif

endif

Variable Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Overall\_Filter)

Variable S=0

SVAR Norm\_point=Norm\_point

If(cmpstr(Norm\_point,"Max")==0)

Wave Norm\_data=T\_Max

else

Wave Norm\_data=Rise\_Midpoint

endif

Variable Start\_P\_longest, End\_P\_longest

Duplicate/O T\_Max, Limit\_L, Limit\_R

Do

If(Peak\_Split1[S])

Limit\_L[S]=x2pnt(Working\_Trace\_Copy,Norm\_data[S])-x2pnt(Working\_Trace\_Copy,Peak\_Split1[S])

else

Limit\_L[S]=x2pnt(Working\_Trace\_Copy,Norm\_data[S])-x2pnt(Working\_Trace\_Copy,T\_Bkg1[S])

endif

Start\_P\_longest=max(Start\_P\_longest,Limit\_L[S])

If(Peak\_Split2[S])

Limit\_R[S]=x2pnt(Working\_Trace\_Copy,Peak\_Split2[S])-x2pnt(Working\_Trace\_Copy,Norm\_data[S])

else

If(S==Total\_peaks\_number-1)

Limit\_R[S]=x2pnt(Working\_Trace\_Copy,T\_Bkg2[S])-x2pnt(Working\_Trace\_Copy,Norm\_data[S])

else

Limit\_R[S]=x2pnt(Working\_Trace\_Copy,min(T\_Bkg2[S], T\_Bkg1[S+1]))-x2pnt(Working\_Trace\_Copy,Norm\_data[S])

endif

endif

End\_P\_longest=max(End\_P\_longest,Limit\_R[S])

S+=1

while(S<Total\_peaks\_number)

Variable Norm\_pnt=x2pnt(Working\_Trace\_Copy,Norm\_data[1])

Duplicate/O/R=[Norm\_pnt-Start\_P\_longest,Norm\_pnt+End\_P\_longest] Working\_trace\_copy, Avg\_peak

SetScale/P x -Start\_P\_longest\*(T\_Delta/1000),(T\_Delta/1000), "s", Avg\_peak

Avg\_peak=0

S=0

Variable Bkg\_level

String Extrap\_Name

Do

Norm\_pnt=x2pnt(Working\_Trace\_Copy,Norm\_data[S])

Duplicate/O/R=[Norm\_pnt-Start\_P\_longest-1,Norm\_pnt+End\_P\_longest] Working\_trace\_copy, Temp\_wave

CopyScales/P Avg\_peak Temp\_wave

Bkg\_level=Temp\_wave[Start\_P\_longest-1-Limit\_L[S]]

Temp\_wave-=Bkg\_level

Temp\_wave[0,Start\_P\_longest-1-Limit\_L[S]]=0

Duplicate/O/R=[0,Start\_P\_longest-1-Limit\_L[S]] Temp\_wave Noizy\_Inset

Noizy\_Inset+=gnoise(Bkg\_noise\_I\*2)

if(numpnts(Noizy\_Inset)>Binomial\_coeff+1)

Smooth Binomial\_coeff, Noizy\_Inset

endif

Temp\_wave[0,Start\_P\_longest-1-Limit\_L[S]]+=Noizy\_Inset(x)

Temp\_wave[Start\_P\_longest-1+Limit\_R[S],numpnts(Temp\_wave)]=0

Duplicate/O/R=[Start\_P\_longest-1+Limit\_R[S],numpnts(Temp\_wave)] Temp\_wave Noizy\_Inset

Noizy\_Inset+=gnoise(Bkg\_noise\_I\*2)

if(numpnts(Noizy\_Inset)>Binomial\_coeff+1)

Smooth Binomial\_coeff, Noizy\_Inset

endif

Temp\_wave[Start\_P\_longest-1+Limit\_R[S],numpnts(Temp\_wave)]+=Noizy\_Inset(x)

Extrap\_Name="Extrap\_"+num2str(S+1)

If(exists(Extrap\_Name)==1)

Duplicate/O $Extrap\_Name, qqq

qqq-=Bkg\_level

SetScale/P x pnt2x(Temp\_wave,(Start\_P\_longest-1+Limit\_R[S])),(T\_Delta/1000), "s", qqq

Temp\_wave[Start\_P\_longest-1+Limit\_R[S],Start\_P\_longest-1+Limit\_R[S]+numpnts(qqq)]+=qqq(x)

endif

CopyScales/P Avg\_peak Temp\_wave

Extrap\_Name="Extrap\_"+num2str(S)

If(exists(Extrap\_Name)==1)

Duplicate/O $Extrap\_Name, qqq

Bkg\_level=max(qqq(0),qqq(numpnts(qqq)))

qqq-=Bkg\_level

CopyScales/P Temp\_wave qqq

Temp\_wave[0,numpnts(qqq)]-=qqq(x)

endif

// string namen="Temp\_wave"+num2str(S)

// Duplicate/O Temp\_wave $namen

Avg\_peak+=Temp\_wave(x)

S+=1

While(S<Total\_peaks\_number)

Avg\_peak/=S

Avg\_Spike\_Weight=S

Killwaves/Z Limit\_L, Limit\_R,qqq

End

Macro Avg\_peak\_graph() : Graph

PauseUpdate; Silent 1

SetDataFolder $"root:Quanta"

Display/K=1/W=(419.25,37.25,579.75,228.5) Avg\_peak as "Averaged Spike"

DoWindow/C Avg\_peak\_graph

ControlBar 45

Button Smooth,pos={1,3},size={50,20},proc=SmoothBtn\_Q,title="Smooth"

SetVariable Smooth\_F,pos={157,5},size={53,16},title=" "

SetVariable Smooth\_F,limits={1,32767,10},value= root:Quanta:Smoothing\_Factor

PopupMenu Smoth\_meth,pos={56,2},size={76,21},proc=Smooth\_method\_Q

PopupMenu Smoth\_meth,mode=3,popvalue="Binomial sm. ",value= #"\"LP Gaussian ;HP Gaussian ;Binomial sm. ;Boxcar sm. ;Sav.-Gol. sm.\""

Button Add\_Avg,pos={1,25},size={65,18},proc=Add\_saved\_avg,title="Add"

Button Add\_Avg,help={"You can sum up averaged peaks from several experiments later!"}

Button Del\_Avg,pos={68,25},size={65,18},proc=Delete\_Avg,title="Clear"

Button Save\_Avg,pos={144,48},size={65,18},proc=Save\_zoomed\_trace,title="Save"

Button Save\_Avg,help={"You can sum up averaged spikes from several experiments later!"}

ValDisplay Avg\_total\_W title="N=",pos={138,26},size={70,15},value=root:Quanta:Avg\_Spike\_Weight

ModifyGraph zero(bottom)=2

AutoPositionWindow/E/M=0/R=File\_Q Avg\_peak\_graph

Endmacro

Function Delete\_Avg(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

Wave Avg\_peak=Avg\_peak

NVAR Avg\_Spike\_Weight=Avg\_Spike\_Weight

DoAlert 1, "Delete averaged spike?"

if(V\_flag==2)

abort

endif

Make/O/N=0 Avg\_peak

Avg\_Spike\_Weight=0

End

Proc Get\_weights(Weight\_Added)

Variable Weight\_Added=gWeight\_Added

Prompt Weight\_Added, "Enter the number of averaged spikes in the Added trace:"

gWeight\_Added=Weight\_Added

End

Function Add\_saved\_avg(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

Variable/G gWeight\_Added

Wave Avg\_peak=Avg\_peak

NVAR Avg\_Spike\_Weight=Avg\_Spike\_Weight

Close /A

variable RefNumber

Open/D/R/T="IGBW" RefNumber

if(strlen(S\_filename)==0)

abort

endif

String Loaded\_wave=S\_fileName

LoadWave/Q/H/O Loaded\_wave

String Added\_Trace\_name=StringFromList(0,S\_waveNames,";")

Execute "Get\_weights()"

Wave Added\_wave=$Added\_Trace\_name

If(numpnts(Avg\_peak)==0)

Duplicate/O Added\_wave Avg\_peak

Avg\_Spike\_Weight+=gWeight\_Added

abort

endif

Avg\_peak\*=Avg\_Spike\_Weight

Added\_wave\*=gWeight\_Added

Variable Org\_DeltaT=pnt2x(Avg\_peak,1)-pnt2x(Avg\_peak,0)

Variable Added\_DeltaT=pnt2x(Added\_wave,1)-pnt2x(Added\_wave,0)

Variable Org\_Npnts=x2pnt(Avg\_peak,0)

Variable Added\_Npnts=x2pnt(Added\_wave,0)

If(Org\_Npnts>Added\_Npnts)

SetScale/P x pnt2x(Avg\_peak,0)+(Org\_Npnts-Added\_Npnts)\*Org\_DeltaT,Org\_DeltaT, "s", Avg\_peak

DeletePoints 0,(Org\_Npnts-Added\_Npnts), Avg\_peak

else

SetScale/P x pnt2x(Added\_wave,0)+(Added\_Npnts-Org\_Npnts)\*Added\_DeltaT,Added\_DeltaT, "s", Added\_wave

DeletePoints 0,(Added\_Npnts-Org\_Npnts), Added\_wave

endif

Org\_Npnts=numpnts(Avg\_peak)-x2pnt(Avg\_peak,0)

Added\_Npnts=numpnts(Added\_wave)-x2pnt(Added\_wave,0)

If(Org\_Npnts>Added\_Npnts)

DeletePoints numpnts(Added\_wave),(Org\_Npnts-Added\_Npnts), Avg\_peak

else

DeletePoints numpnts(Avg\_peak),(Added\_Npnts-Org\_Npnts), Added\_wave

endif

Avg\_peak+=Added\_wave

Avg\_peak/=(Avg\_Spike\_Weight+gWeight\_Added)

Avg\_Spike\_Weight+=gWeight\_Added

KillWaves/Z Added\_wave

End

Function Killer\_of\_Hookers(infoStr)

String infoStr

Variable somethingDone=0

String win = StringByKey("WINDOW",infoStr)

String event = StringByKey("EVENT",infoStr)

if (CmpStr(event, "kill") != 0)

return 0

endif

if (CmpStr(win, "Avg\_peak\_graph") != 0)

DoWindow/K Avg\_peak\_graph

somethingDone=1

endif

if (CmpStr(win, "Avg\_peak\_pnts") != 0)

DoWindow/K Avg\_peak\_pnts

somethingDone=1

endif

return somethingDone

end

//\_\_\_\_\_\_\_\_\_\_Stats Notebook\_\_\_\_\_\_\_\_\_\_\_\_

Function Show\_stats(ctrlName) : ButtonControl

String ctrlName

if (CheckName("Stats", 10)==0)

NewNotebook/N=Stats/F=1/V=1/W=(2.4,121.4,604.2,305) as "Data analysis"

Generate\_notebook()

else

DoWindow /F Stats

MoveWindow 1,1,1,1

Notebook Stats selection={startOfFile, endOfFile}

Generate\_notebook()

endif

End

Function Generate\_notebook()

SetDataFolder $"root:Quanta"

Wave T\_Max=T\_Max

SVAR Loaded\_file\_path=Loaded\_file\_path

String Row\_to\_Print

String/G One\_wave

String Stat\_result

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Overall\_Filter=Overall\_Filter

NVAR Bkg\_noise\_I=Bkg\_noise\_I

NVAR Bkg\_noise\_dI=Bkg\_noise\_dI

NVAR Detection\_Mult=Detection\_Mult

NVAR Detection\_Foot\_Mult=Detection\_Foot\_Mult

NVAR Smoothing\_Factor\_Add=Smoothing\_Factor\_Add

NVAR Smoothing\_Factor\_diff1=Smoothing\_Factor\_diff1

NVAR Smooth\_more=Smooth\_more

NVAR Smooth\_Derivative=Smooth\_Derivative

NVAR Spike\_Min\_Imax=Spike\_Min\_Imax

NVAR Spike\_Max\_T05=Spike\_Max\_T05

NVAR Spike\_Max\_Trise=Spike\_Max\_Trise

NVAR Foot\_Min\_W=Foot\_Min\_W

NVAR Foot\_Min\_H=Foot\_Min\_H

NVAR SSFoot\_Do=SSFoot\_Do

NVAR Native\_Foot\_Del=Native\_Foot\_Del

NVAR Baseline\_Drift=Baseline\_Drift

NVAR Overlap\_Prc=Overlap\_Prc

SVAR Overlaps=Overlaps

SVAR Values\_to\_show=Values\_to\_show

SVAR Stats\_names=Stats\_names

SVAR Population\_Center=Population\_Center

String nb = "Stats"

Notebook $nb defaultTab=36, statusWidth=238, pageMargins={72,72,72,72}

Notebook $nb showRuler=1, rulerUnits=1, updating={1, 3600}

Notebook $nb newRuler=Normal, justification=0, margins={0,0,468}, spacing={0,0,0}, tabs={}, rulerDefaults={"Arial",7,0,(0,0,0)}

Notebook $nb ruler=Normal, specialChar={3,0,""}

Notebook $nb text="\r"

Notebook $nb text=Loaded\_file\_path+"\r"

Notebook $nb text="Trace filtering (-3dB Gaussian):\r"

Notebook $nb text="\tCurrent trace - "+num2str(Overall\_Filter)+" Hz\r"

If(Smooth\_more)

Notebook $nb text="\tAdditional trace filtering - "+num2str(Smoothing\_Factor\_Add)+" Hz\r"

endif

If(Smooth\_Derivative)

Notebook $nb text="\tDifferentiated trace filtering - "+num2str(Smoothing\_Factor\_diff1)+" Hz\r"

endif

Notebook $nb text="Noise level: SD(I) = "+num2str(Bkg\_noise\_I)+"; SD(dI/dt) = "+num2str(Bkg\_noise\_dI)

Notebook $nb text="\rDetection threshold: Spikes - SD(dI/dt)\*"+num2str(Detection\_Mult)+"; Foot - SD(I)\*"+num2str(Detection\_Foot\_Mult)

Notebook $nb text="\rCutoffs used:\r"

If(Spike\_Min\_Imax)

Notebook $nb text="\tMin I(max) - "+num2str(Spike\_Min\_Imax)+" pA\r"

endif

If(Spike\_Max\_T05)

Notebook $nb text="\tMax T(1/2) - "+num2str(Spike\_Max\_T05)+" ms\r"

endif

If(Spike\_Max\_Trise)

Notebook $nb text="\tMax T(rise) - "+num2str(Spike\_Max\_Trise)+" ms\r"

endif

If(Foot\_Min\_H)

Notebook $nb text="\tMin I(foot) - "+num2str(Foot\_Min\_H)+" pA\r"

endif

If(SSFoot\_Do)

Notebook $nb text="\tOnly PSF with steady states longer than "+num2str(Foot\_Min\_W)+" ms were analized\r"

else

If(Foot\_Min\_W)

Notebook $nb text="\tMin T(foot) - "+num2str(Foot\_Min\_W)+" ms\r"

endif

endif

If(Native\_Foot\_Del)

Notebook $nb text="\t'Native' PSF were deleted\r"

endif

Notebook $nb text="\tAllowed baseline drift - "+num2str(Baseline\_Drift)+"%\r"

Notebook $nb text="\tAllowed maximal overlap - "+num2str(Overlap\_Prc)+"%\r"

Notebook $nb text="\tThe remaining overlaps - "+Overlaps

Notebook $nb text="\r\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

Notebook $nb text="\r\rFound "+num2str(Total\_peaks\_number)+" events\r"

Notebook $nb ruler=Normal; Notebook $nb margins={0,0,720}, rulerDefaults={"Arial",7,1,(0,0,0)}, tabs={36,72,108,144,180,216,252,288,324,360,396,432,468,504,541,576}

Notebook $nb text=Stats\_names

If(Total\_peaks\_number==0)

abort

endif

String Waves\_to\_show="Interspike\_interval,"+Values\_to\_show

Make/O/N=(Total\_peaks\_number-1) Interspike\_interval

Variable i=0

do

If(i>0)

Interspike\_interval[i-1]=(T\_Max[i]-T\_Max[i-1])\*1000

endif

i+=1

while (i<Total\_peaks\_number)

Variable ii=0

String Mean\_line

If(cmpstr(Population\_Center, "Mean" )==0)

Mean\_line="Mean\t"

else

Mean\_line="Median\t"

endif

String SD\_line="SD\t"

String SE\_line="SE\t"

String N\_line="N\t"

Stat\_result=Mean\_SD("Interspike\_interval")

Mean\_line+=StringFromList(0, Stat\_result ,";")+"\t\t"

SD\_line+=StringFromList(1, Stat\_result ,";")+"\t\t"

SE\_line+=StringFromList(2, Stat\_result ,";")+"\t\t"

N\_line+=StringFromList(3, Stat\_result ,";")+"\t\t"

ii=2

Variable exclude

do

exclude=0

One\_wave=StringFromList(ii, Waves\_to\_show ,",")

if( strlen(One\_wave) == 0 )

break

endif

If((cmpstr(One\_wave,"T\_Max")==0)%|(cmpstr(One\_wave,"T\_Bkg1")==0))

exclude=1

else

If((cmpstr(One\_wave,"T\_Bkg2")==0)%|(cmpstr(One\_wave,"Fall\_fit")==0))

exclude=1

endif

endif

If(exclude==1)

Mean\_line+="\t"

SD\_line+="\t"

SE\_line+="\t"

N\_line+="\t"

else

Stat\_result=Mean\_SD(One\_wave)

Mean\_line+=StringFromList(0, Stat\_result ,";")+"\t"

SD\_line+=StringFromList(1, Stat\_result ,";")+"\t"

SE\_line+=StringFromList(2, Stat\_result ,";")+"\t"

N\_line+=StringFromList(3, Stat\_result ,";")+"\t"

endif

ii+=1

while(ii)

InsertPoints 0,1, Interspike\_interval

Notebook $nb text=Mean\_line+"\r"

Notebook $nb text=SD\_line+"\r"

Notebook $nb text=SE\_line+"\r"

Notebook $nb text=N\_line+"\r"

Notebook $nb text="\r"

Notebook $nb text=Stats\_names

i=0

do

Row\_to\_Print=""

ii=0

do

if(strlen(StringFromList(ii, Waves\_to\_show ,",")) == 0 )

break

endif

If(cmpstr(StringFromList(ii, Waves\_to\_show ,","),"Fall\_fit")==0)

Wave/T Fall\_fit=Fall\_fit

Row\_to\_Print+=Fall\_fit[i]+"\t"

else

One\_wave="One\_wave="+"num2str("+StringFromList(ii, Waves\_to\_show ,",")+"["+num2str(i)+"])"

execute/Z One\_wave

Row\_to\_Print+=One\_wave+"\t"

endif

ii+=1

while(ii)

Row\_to\_Print+="\r"

Notebook $nb text="\t"+Row\_to\_Print

i+=1

while (i<Total\_peaks\_number)

Notebook $nb selection={startOfFile, endOfFile}

return 0

End

Function/S Mean\_SD(name)

String name

SetDataFolder $"root:Quanta"

SVAR Population\_Center=Population\_Center

String Stat\_result

Duplicate/o $name, Stat\_wave

Sort Stat\_wave Stat\_wave

Do

If (Stat\_wave[0]<0.0000001) // the smallest possible number to consider. Change if nassesary.

Deletepoints 0, 1, Stat\_wave

else

break

endif

While (1)

If(numpnts(Stat\_wave)<=1)

Stat\_result=" ; ; ; ; ;"

else

wavestats/Q Stat\_wave

If(cmpstr(Population\_Center, "Mean" )==0)

Stat\_result=num2str(V\_avg)+";"

else

SetScale/P x 0,1,Stat\_wave

Variable Median = Stat\_wave((numpnts(Stat\_wave)-1)/2)

Stat\_result=num2str(Median)+";"

endif

Stat\_result+=num2str(V\_sdev)+";"+num2str(V\_sdev/sqrt(V\_npnts))+";"+num2str(V\_npnts)

endif

return Stat\_result

end

//\_\_\_\_\_\_\_\_\_Detection Limits\_\_\_\_\_\_\_\_\_\_\_

// Returns 1 if a spike has invalide shape, and 2 if it does not pass a cutoff.

Function Detection\_limits(Peak\_pnt)

Variable Peak\_pnt

SetDataFolder $"root:Quanta"

NVAR Spike\_Max\_T05=Spike\_Max\_T05

NVAR Spike\_Max\_Trise=Spike\_Max\_Trise

NVAR Spike\_Min\_Imax=Spike\_Min\_Imax

Wave Peak\_Q=Peak\_Q

Wave Peak\_Imax=Peak\_Imax

Wave Peak\_t05=Peak\_t05

Wave Fall\_slope=Fall\_slope

Wave Rise\_slope=Rise\_slope

Wave Rise\_time=Rise\_time

If ((Peak\_Q[Peak\_pnt]<=0)%|(Peak\_t05[Peak\_pnt]<=0))

return 1

endif

If ((Fall\_slope[Peak\_pnt]<=0)%|(Rise\_slope[Peak\_pnt]<=0))

return 1

endif

If ((Spike\_Max\_Trise>0)&(Rise\_time[Peak\_pnt]>Spike\_Max\_Trise))

return 2

endif

If ((Spike\_Max\_T05>0)&(Peak\_t05[Peak\_pnt]>Spike\_Max\_T05))

return 2

endif

If (Peak\_Imax[Peak\_pnt]<Spike\_Min\_Imax)

return 2

endif

return 0

End

Function Detection\_limits\_Foot(Peak\_pnt)

Variable Peak\_pnt

SetDataFolder $"root:Quanta"

NVAR Foot\_Min\_W=Foot\_Min\_W

NVAR Foot\_Min\_H=Foot\_Min\_H

Wave Foot\_Q=Foot\_Q

Wave Foot\_I=Foot\_I

Wave Foot\_W=Foot\_W

If ((Foot\_W[Peak\_pnt]<=Foot\_Min\_W)%|(Foot\_I[Peak\_pnt]<=Foot\_Min\_H))

return 1

endif

if(Foot\_Q[Peak\_pnt]<=0)

return 1

endif

Wavestats/Q Foot\_W

If(V\_numNans!=0)

return 1

endif

return 0

End

Function Change\_Detection\_Limits(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

SetDataFolder $"root:Quanta"

NVAR Spike\_Min\_Imax=Spike\_Min\_Imax

NVAR Spike\_Min\_Imax\_Last=Spike\_Min\_Imax\_Last

NVAR Spike\_Max\_T05=Spike\_Max\_T05

NVAR Spike\_Max\_T05\_Last=Spike\_Max\_T05\_Last

NVAR Spike\_Max\_Trise=Spike\_Max\_Trise

NVAR Spike\_Max\_Trise\_Last=Spike\_Max\_Trise\_Last

NVAR Foot\_Min\_H=Foot\_Min\_H

NVAR Foot\_Min\_H\_Last=Foot\_Min\_H\_Last

strswitch(ctrlName)

case "Detection\_Spike\_Imax":

If(checked==1)

Spike\_Min\_Imax=Spike\_Min\_Imax\_Last

else

Spike\_Min\_Imax\_Last=Spike\_Min\_Imax

Spike\_Min\_Imax=0

endif

break

case "Detection\_Spike\_T05":

If(checked==1)

Spike\_Max\_T05=Spike\_Max\_T05\_Last

else

Spike\_Max\_T05\_Last=Spike\_Max\_T05

Spike\_Max\_T05=0

endif

break

case "Detection\_Spike\_Trise":

If(checked==1)

Spike\_Max\_Trise=Spike\_Max\_Trise\_Last

else

Spike\_Max\_Trise\_Last=Spike\_Max\_Trise

Spike\_Max\_Trise=0

endif

break

case "Detection\_Foot\_H":

If(checked==1)

Foot\_Min\_H=Foot\_Min\_H\_Last

else

Foot\_Min\_H\_Last=Foot\_Min\_H

Foot\_Min\_H=0

endif

break

endswitch

End

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_\_\_\_\_\_Other controls\_\_\_\_\_\_\_\_\_\_\_

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Function Foot\_Min\_W\_chk(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

SetDataFolder $"root:Quanta"

NVAR Foot\_Min\_W=Foot\_Min\_W

NVAR Foot\_Min\_W\_Last=Foot\_Min\_W\_Last

NVAR SSFoot\_Do=SSFoot\_Do

NVAR Native\_Foot\_Del=Native\_Foot\_Del

strswitch (ctrlName)

case "Detection\_Foot\_W":

If(checked==1)

Foot\_Min\_W=Foot\_Min\_W\_Last

else

CheckBox Detection\_SSFoot\_Chk,value= 0

Foot\_Min\_W\_Last=Foot\_Min\_W

Foot\_Min\_W=0

SSFoot\_Do=0

endif

break

case "Detection\_SSFoot\_Chk":

If(checked==1)

CheckBox Detection\_Foot\_W,value= 1

Foot\_Min\_W=Foot\_Min\_W\_Last

SSFoot\_Do=1

else

CheckBox Detection\_Foot\_W,value= 0

Foot\_Min\_W\_Last=Foot\_Min\_W

Foot\_Min\_W=0

SSFoot\_Do=0

endif

break

case "Detection\_Native\_Foot\_Chk":

If(checked==1)

Native\_Foot\_Del=1

else

Native\_Foot\_Del=0

endif

break

endswitch

End

Function Check\_the\_Box(ctrlName,varNum,varStr,varName) : SetVariableControl

String ctrlName

Variable varNum

String varStr

String varName

NVAR Spike\_Min\_Imax=Spike\_Min\_Imax

NVAR Spike\_Max\_T05=Spike\_Max\_T05

NVAR Spike\_Max\_Trise=Spike\_Max\_Trise

NVAR Foot\_Min\_W=Foot\_Min\_W

NVAR Foot\_Min\_H=Foot\_Min\_H

NVAR Smooth\_more=Smooth\_more

NVAR Smooth\_Derivative=Smooth\_Derivative

strswitch (varName)

case "Smoothing\_Factor\_Add":

Smooth\_more=1

CheckBox Scales\_Smooth\_Add,value=1

break

case "Smoothing\_Factor\_diff1":

Smooth\_Derivative=1

CheckBox Scales\_Smooth\_Diff,value=1

break

case "Spike\_Min\_Imax":

If(Spike\_Min\_Imax>0)

CheckBox Detection\_Spike\_Imax,value=1

else

CheckBox Detection\_Spike\_Imax,value=0

endif

break

case "Spike\_Max\_T05":

If(Spike\_Max\_T05>0)

CheckBox Detection\_Spike\_t05,value=1

else

CheckBox Detection\_Spike\_t05,value=0

endif

break

case "Spike\_Max\_Trise":

If(Spike\_Max\_Trise>0)

CheckBox Detection\_Spike\_Trise,value=1

else

CheckBox Detection\_Spike\_Trise,value=0

endif

break

case "Foot\_Min\_H":

If(Foot\_Min\_H>0)

CheckBox Detection\_Foot\_H,value=1

else

CheckBox Detection\_Foot\_H,value=0

endif

break

case "Foot\_Min\_W":

If(Foot\_Min\_W>0)

CheckBox Detection\_Foot\_W,value=1

else

CheckBox Detection\_Foot\_W,value=0

CheckBox Detection\_SSFoot\_Chk,value= 0

endif

break

endswitch

End

Function Close\_Options(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

String Win\_name=WinName(0,64)

NVAR T\_Start=T\_Start

NVAR T\_End=T\_End

WAVE Working\_trace\_copy=Working\_trace\_copy

NVAR T\_Delta=T\_Delta

NVAR Gain=Gain

NVAR Gain\_Temp=Gain\_Temp

T\_Start=pnt2x(Working\_trace\_copy,0)

T\_End=pnt2x(Working\_trace\_copy,(numpnts(Working\_trace\_copy)-1))

T\_Delta=(pnt2x(Working\_trace\_copy,1)-pnt2x(Working\_trace\_copy,0))\*1000

Gain\_Temp=Gain

Dowindow/K $Win\_name

End

Function Show\_Extras(theTag) : ButtonControl

String theTag

SetDataFolder $"root:Quanta"

NVAR Peak\_ID=Peak\_ID

NVAR Show\_Legend=Show\_Legend

NVAR Total\_peaks\_number=Total\_peaks\_number

Variable Peak\_pnt=Peak\_ID-1

strswitch (theTag)

case "Zoom\_Off":

Button Zoom\_Off rename=Zoom\_On, title="Hide Zoom"

// MoveWindow /W=Main\_window 3,80,549,394.25

execute "Zoom\_Win()"

AutoPositionWindow/E/M=0/R=Main\_window Zoom\_Win

SetVariable ID,limits={1,(Total\_peaks\_number),1},win=Zoom\_Win

If(Peak\_ID)

Draw\_lines\_zoom\_window(Peak\_pnt)

endif

Dowindow/F Zoom\_Win

break

case "Zoom\_On":

Button Zoom\_On rename=Zoom\_Off, title="Zoom Win"

Dowindow/K Zoom\_Win

// MoveWindow /W=Main\_window 3,80,762,394.25

break

case "Legend\_Off":

Show\_Legend=1

Button Legend\_Off rename=Legend\_On, title="Hide Legend"

Generate\_annotation(Peak\_pnt)

break

case "Legend\_On":

Show\_Legend=0

TextBox/W=Main\_window/N=Peak\_data/K

Button Legend\_On rename=Legend\_Off, title="Show Legend"

break

endswitch

End

Function Show\_options\_panel(ctrlName) : ButtonControl

String ctrlName

if (CheckName("Options\_Tab\_Panels", 6)==0)

execute "Options\_Tab\_Panels(0)"

else

Dowindow/F Options\_Tab\_Panels

endif

strswitch (ctrlName)

case "Menu\_FiltersScales":

Redraw\_Tabs(ctrlName,0)

break

case "Menu\_Cutoffs":

TabControl Tab\_thing, value= 1

Redraw\_Tabs(ctrlName,1)

break

case "Menu\_Results":

TabControl Tab\_thing, value= 2

Redraw\_Tabs(ctrlName,2)

break

case "Menu\_Stats":

TabControl Tab\_thing, value= 3

Redraw\_Tabs(ctrlName,3)

break

endswitch

End

Function Redraw\_Tabs(name,tabNumber)

String name

Variable tabNumber

SetDataFolder $"root:Quanta"

String Existing\_controls=ControlNameList("")

NVAR T\_Start=T\_Start

NVAR T\_End=T\_End

WAVE Working\_trace\_copy=Working\_trace\_copy

NVAR T\_Delta=T\_Delta

NVAR Gain=Gain

NVAR Gain\_Temp=Gain\_Temp

T\_Start=pnt2x(Working\_trace\_copy,0)

T\_End=pnt2x(Working\_trace\_copy,(numpnts(Working\_trace\_copy)-1))

T\_Delta=(pnt2x(Working\_trace\_copy,1)-pnt2x(Working\_trace\_copy,0))\*1000

Gain\_Temp=Gain

Options\_Tab\_Panels\_controls(tabNumber)

Button Recalculate\_btn,win=Options\_Tab\_Panels, fColor=(0,0,0)

If(tabNumber==0)

NVAR Smooth\_more=Smooth\_more

If(Smooth\_more==1)

CheckBox Scales\_Smooth\_Add,value=1

endif

NVAR Smooth\_Derivative=Smooth\_Derivative

If(Smooth\_Derivative==1)

CheckBox Scales\_Smooth\_Diff,value=1

endif

endif

If(tabNumber==1)

NVAR Spike\_Min\_Imax=Spike\_Min\_Imax

NVAR Spike\_Max\_T05=Spike\_Max\_T05

NVAR Spike\_Max\_Trise=Spike\_Max\_Trise

NVAR Foot\_Min\_W=Foot\_Min\_W

NVAR SSFoot\_Do=SSFoot\_Do

NVAR Native\_Foot\_Del=Native\_Foot\_Del

NVAR Foot\_Min\_H=Foot\_Min\_H

SVAR Overlaps=Overlaps

If(Spike\_Min\_Imax>0)

CheckBox Detection\_Spike\_Imax,value=1

endif

If(Spike\_Max\_T05>0)

CheckBox Detection\_Spike\_t05,value=1

endif

If(Spike\_Max\_Trise>0)

CheckBox Detection\_Spike\_Trise,value=1

endif

If(Foot\_Min\_H>0)

CheckBox Detection\_Foot\_H,value=1

endif

If(Foot\_Min\_W>0)

CheckBox Detection\_Foot\_W,value=1

endif

If(SSFoot\_Do==1)

CheckBox Detection\_SSFoot\_Chk,value=1

endif

If(Native\_Foot\_Del==1)

CheckBox Detection\_Native\_Foot\_Chk,value=1

endif

If(cmpstr(Overlaps,"Ignore")==0)

PopupMenu Detection\_Overlaps,mode=1, win=Options\_Tab\_Panels

else

If(cmpstr(Overlaps,"Separate")==0)

PopupMenu Detection\_Overlaps,mode=2, win=Options\_Tab\_Panels

else

PopupMenu Detection\_Overlaps,mode=3, win=Options\_Tab\_Panels

endif

endif

endif

If(tabNumber==2)

NVAR Show\_Time=Show\_Time

NVAR Show\_Base=Show\_Base

NVAR Show\_Width=Show\_Width

NVAR Show\_H=Show\_H

NVAR Show\_Q=Show\_Q

NVAR Show\_Molec=Show\_Molec

NVAR Show\_Rise\_t=Show\_Rise\_t

NVAR Show\_Rise\_r=Show\_Rise\_r

NVAR Show\_Fall\_t=Show\_Fall\_t

NVAR Show\_Fall\_r=Show\_Fall\_r

NVAR Show\_Ft\_H=Show\_Ft\_H

NVAR Show\_Ft\_width=Show\_Ft\_width

NVAR Show\_Ft\_Q=Show\_Ft\_Q

NVAR Show\_Ft\_molec=Show\_Ft\_molec

SVAR Fit\_method=Fit\_method

SetDrawEnv fname= "Arial"

DrawText 115,191,"%"

If(Show\_Time==1)

CheckBox Results\_Show1,value=1

endif

If(Show\_Base==1)

CheckBox Results\_Show2,value=1

endif

If(Show\_Width==1)

CheckBox Results\_Show4,value=1

endif

If(Show\_H==1)

CheckBox Results\_Show5,value=1

endif

If(Show\_Q==1)

CheckBox Results\_Show6,value=1

endif

If(Show\_Molec==1)

CheckBox Results\_Show7,value=1

endif

If(Show\_Rise\_t==1)

CheckBox Results\_Show8,value=1

endif

If(Show\_Rise\_r==1)

CheckBox Results\_Show9,value=1

endif

If(Show\_Fall\_t==1)

CheckBox Results\_Show10,value=1

endif

If(Show\_Fall\_r==1)

CheckBox Results\_Show11,value=1

endif

If(Show\_Ft\_H==1)

CheckBox Results\_Show12,value=1

endif

If(Show\_Ft\_width==1)

CheckBox Results\_Show13,value=1

endif

If(Show\_Ft\_Q==1)

CheckBox Results\_Show14,value=1

endif

If(Show\_Ft\_molec==1)

CheckBox Results\_Show15,value=1

endif

If(cmpstr(Fit\_method,"Line")==0)

PopupMenu Results\_Fallfit,mode=1,popvalue="Line", win=Options\_Tab\_Panels

else

If(cmpstr(Fit\_method,"Exp")==0)

PopupMenu Results\_Fallfit,mode=2,popvalue="Exp", win=Options\_Tab\_Panels

else

PopupMenu Results\_Fallfit,mode=3,popvalue="DblExp", win=Options\_Tab\_Panels

SetVariable Results\_Fall\_Chi, win=Options\_Tab\_Panels, disable=0

endif

endif

endif

If(tabNumber==3)

SVAR Population\_Center=Population\_Center

SVAR Norm\_point=Norm\_point

If(cmpstr(Population\_Center,"Mean")==0)

PopupMenu Stats\_Population,mode=2,popvalue="Mean", win=Options\_Tab\_Panels

endif

If(cmpstr(Norm\_point,"Rise")==0)

PopupMenu Stats\_AvePeak,mode=2,popvalue="Rise", win=Options\_Tab\_Panels

endif

endif

end

Function Overlaps\_PopMenu(theTag,popNum,popStr) : PopupMenuControl

String theTag

Variable popNum

String popStr

SetDataFolder $"root:Quanta"

SVAR Overlaps=Overlaps

Overlaps=popStr

Redraw\_Tabs(theTag,1)

End

Function Scales\_Switch(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

SetDataFolder $"root:Quanta"

Variable Radio\_On=1

strswitch (ctrlName)

case "Scales\_Change\_Delta":

Radio\_On= 1

SetVariable Scales\_End\_time,disable=2

SetVariable Scales\_Delta\_Time,disable=0

break

case "Scales\_Change\_End":

Radio\_On= 2

SetVariable Scales\_End\_time,disable=0

SetVariable Scales\_Delta\_Time,disable=2

break

endswitch

CheckBox Scales\_Change\_Delta,mode=1, value= Radio\_On==1

CheckBox Scales\_Change\_End,mode=1, value= Radio\_On==2

End

Window Options\_Tab\_Panels(Tab) : Panel

Variable Tab

PauseUpdate; Silent 1

NewPanel /K=1 /W=(235,104,506,401) as "Detection Options"

TabControl Tab\_thing,pos={6,6},size={260,261},proc=Redraw\_Tabs

TabControl Tab\_thing,tabLabel(0)="Filters/Scales",tabLabel(1)="Cutoffs",tabLabel(2)="Results",tabLabel(3)="Stats"

TabControl Tab\_thing,value= tab

End

Function Options\_Tab\_Panels\_controls(Tab)

Variable Tab

SetDrawLayer/K UserBack

NVAR T\_Delta=T\_Delta

Button Close\_window,pos={65,273},size={70,20},proc=Close\_Options,title="Close"

Button Recalculate\_btn,pos={138,273},size={70,20},proc=Recalculate\_Peaks\_Btn,title="Recalc"

Variable Min\_Freq=Binomial\_to\_Gaussian\_Calc(32767)

GroupBox Scales\_Smooth\_Box,pos={18,27},size={235,85},fColor=(26112,0,10240), title="Filters",disable= (tab!=0)

Button Scales\_Estimate\_Filters,pos={80,46},size={109,18},proc=Filter\_estimate,title="Estimate Filters",disable=(tab!=0)

Button Scales\_Estimate\_Filters,help={"Estimates the corner frequencies of the three filters used during trace preconditioning."}

CheckBox Scales\_Smooth\_Add,pos={25,70},size={124,14},proc=Additional\_Filters,title="Filter trace additionally"

CheckBox Scales\_Smooth\_Add,value= 0,disable= (tab!=0)

CheckBox Scales\_Smooth\_Add,help={"Additional trace filtering (Gaussian). Increases signal-to-noise, but does not affect spike parameters."}

CheckBox Scales\_Smooth\_Diff,pos={25,90},size={124,14},proc=Additional\_Filters,title="Filter the 1st derivative"

CheckBox Scales\_Smooth\_Diff,value= 0,disable= (tab!=0)

CheckBox Scales\_Smooth\_Diff,help={"Gaussian filtering of differentiated trace. Increases signal-to-noise, but does not affect spike parameters."}

SetVariable Scales\_Smooth\_F,pos={156,70},size={84,16},title=" ",format="%.1W1PHz", proc=Check\_the\_Box,disable= (tab!=0)

SetVariable Scales\_Smooth\_F,limits={Min\_Freq+1,(1/(T\_Delta/1000))/2,50},value= root:Quanta:Smoothing\_Factor\_Add

SetVariable Scales\_Diff\_smooth,pos={156,90},size={84,16},title=" ",format="%.1W1PHz", proc=Check\_the\_Box,disable= (tab!=0)

SetVariable Scales\_Diff\_smooth,limits={Min\_Freq+1,(1/(T\_Delta/1000))/2,50},value= root:Quanta:Smoothing\_Factor\_diff1

GroupBox Scales\_Y\_Box,pos={18,116},size={235,40},title="Y scale (pA)",fColor=(26112,0,10240),disable= (tab!=0)

SetVariable Scales\_Mult,pos={47,134},size={193,16},title="Multiply the current by:",disable= (tab!=0)

SetVariable Scales\_Mult,fSize=10,limits={0,1e+18,10},value= root:Quanta:Gain\_Temp,proc=Adjust\_Trace\_Scales

SetVariable Scales\_Mult,help={"Gain. The current has to be in pA!"}

GroupBox Scales\_X\_Box,pos={18,160},size={235,98}, title="X scale (seconds)",disable= (tab!=0),fColor=(26112,0,10240),disable= (tab!=0)

SetVariable Scales\_Start\_time,pos={54,181},size={136,16},title="Start (s)",fSize=10,disable= (tab!=0)

SetVariable Scales\_Start\_time,limits={0,Inf,1},value= root:Quanta:T\_Start,proc=Adjust\_Trace\_Scales

SetVariable Scales\_Start\_time,help={"Set the time of the first datapoint (seconds)."}

SetVariable Scales\_End\_time,pos={57,203},size={133,16},fSize=10,title="End (s)",disable= (tab!=0)

SetVariable Scales\_End\_time,help={"Set the time of the last datapoint (seconds)."}

SetVariable Scales\_End\_time,limits={0,Inf,1},value= root:Quanta:T\_End,proc=Adjust\_Trace\_Scales

SetVariable Scales\_Delta\_Time,pos={43,224},size={147,16},fSize=10,fstyle=0, title="Delta (ms)",disable= (tab!=0),format="%g"

SetVariable Scales\_Delta\_Time,limits={0.001,Inf,0.01},value= root:Quanta:T\_Delta,proc=Adjust\_Trace\_Scales

SetVariable Scales\_Delta\_Time,help={"Set the sampling interval (milli-seconds)."}

ValDisplay Scales\_Delta\_Hz,pos={80,240},size={74,16},fSize=10,fstyle=0,format="%.1W1PHz",mode=2

ValDisplay Scales\_Delta\_Hz,limits={0,0,0},barmisc={0,1000},disable= (tab!=0)

ValDisplay Scales\_Delta\_Hz,help={"Sampling frequency."},value= #"1/(root:Quanta:T\_Delta/1000)"

CheckBox Scales\_Change\_End,pos={198,205},size={109,16},proc=Scales\_Switch,title=" ",mode=1,value= 0,disable= (tab!=0)

CheckBox Scales\_Change\_Delta,pos={198,226},size={103,16},proc=Scales\_Switch,title=" ",mode=1,value= 1,disable= (tab!=0)

If(tab==0)

SetVariable Scales\_End\_Time,disable=2

Button Recalculate\_btn,proc=Change\_trace\_scales,title="Change"

endif

GroupBox Detection\_Box\_Spike,pos={18,27},size={235,77},fColor=(26112,0,10240), title="Spike cutoffs",disable= (tab!=1)

CheckBox Detection\_Spike\_Imax,pos={31,45},size={97,14},proc=Change\_Detection\_Limits,title="Min Spike I(max), (pA)"

CheckBox Detection\_Spike\_Imax,value= 0,disable= (tab!=1)

SetVariable Detection\_Set\_Spike\_Imax,pos={163,45},size={75,16},title=" ", proc=Check\_the\_Box

SetVariable Detection\_Set\_Spike\_Imax,limits={0,Inf,1},value= root:Quanta:Spike\_Min\_Imax,disable= (tab!=1)

CheckBox Detection\_Spike\_t05,pos={31,64},size={100,14},proc=Change\_Detection\_Limits,title="Max Spike T(1/2), (ms)"

CheckBox Detection\_Spike\_t05,value= 0,disable= (tab!=1)

SetVariable Detection\_Set\_Spike\_t05,pos={163,64},size={75,16},title=" ", proc=Check\_the\_Box

SetVariable Detection\_Set\_Spike\_t05,limits={0,Inf,0.5},value= root:Quanta:Spike\_Max\_T05,disable= (tab!=1)

CheckBox Detection\_Spike\_trise,pos={31,83},size={100,14},proc=Change\_Detection\_Limits,title="Max Spike T(rise), (ms)"

CheckBox Detection\_Spike\_trise,value= 0,disable= (tab!=1)

SetVariable Detection\_Set\_Spike\_trise,pos={163,83},size={75,16},title=" ", proc=Check\_the\_Box

SetVariable Detection\_Set\_Spike\_trise,limits={0,Inf,0.5},value= root:Quanta:Spike\_Max\_Trise,disable= (tab!=1)

GroupBox Detection\_Box\_Foot,pos={18,106},size={235,73},fColor=(26112,0,10240), title="Foot cutoffs",disable= (tab!=1)

CheckBox Detection\_Foot\_H,pos={31,122},size={90,14},proc=Change\_Detection\_Limits,title="Min Foot Height (pA)"

CheckBox Detection\_Foot\_H,value= 0,disable= (tab!=1)

SetVariable Detection\_Set\_Min\_Foot\_H,pos={163,123},size={75,16},title=" ", proc=Check\_the\_Box

SetVariable Detection\_Set\_Min\_Foot\_H,limits={0,Inf,1},value= root:Quanta:Foot\_Min\_H,disable= (tab!=1)

CheckBox Detection\_Foot\_W,pos={31,141},size={96,14},proc=Foot\_Min\_W\_chk,title="Min Foot Width (ms)"

CheckBox Detection\_Foot\_W,value= 0,disable= (tab!=1)

SetVariable Detection\_Set\_Min\_Foot\_W,pos={163,141},size={75,16},title=" ", proc=Check\_the\_Box

SetVariable Detection\_Set\_Min\_Foot\_W,limits={0,Inf,0.5},value= root:Quanta:Foot\_Min\_W,disable= (tab!=1)

CheckBox Detection\_SSFoot\_Chk,pos={31,160},size={96,14},proc=Foot\_Min\_W\_chk,title="Steady-state feet only"

CheckBox Detection\_SSFoot\_Chk,value= 0,disable= (tab!=1)

CheckBox Detection\_SSFoot\_Chk, help={"Only feet with steady-states that persist for 'Min Foot Width' duration will be considered."}

CheckBox Detection\_Native\_Foot\_Chk,pos={162,160},size={96,14},proc=Foot\_Min\_W\_chk,title="Delete 'native'"

CheckBox Detection\_Native\_Foot\_Chk,value= 0,disable= (tab!=1)

CheckBox Detection\_Native\_Foot\_Chk, help={"'Native' foot width=0.33\*(50-90%)risetime. (Chow, 95)."}

GroupBox Detection\_Bad\_Peaks,pos={18,180},size={235,78},fColor=(26112,0,10240), title="'Bad' spikes",disable= (tab!=1)

SetVariable Detection\_Baseline\_Drift,pos={71,197},size={167,18},title="Allowed Baseline Drift",format="%.0W0P%"

SetVariable Detection\_Baseline\_Drift,limits={1,100,5},value= root:Quanta:Baseline\_Drift,disable= (tab!=1)

SetVariable Detection\_Baseline\_Drift, help={"Baseline drift is the delta current between spike beginning and end."}

SetVariable Detection\_Overlap\_Prc,pos={33,216},size={205,16},title="Delete spike overlapping by >",format="%.0W0P%"

SetVariable Detection\_Overlap\_Prc,limits={0,100,5},value= root:Quanta:Overlap\_Prc,disable= (tab!=1)

SetVariable Detection\_Overlap\_Prc, help={"% overlap is the ratio of the current at the minimum between two spikes and the I(max) of the smaller spike."}

PopupMenu Detection\_Overlaps,pos={57,234},proc=Overlaps\_PopMenu,title="Remaining Overlaps:",disable= (tab!=1)

PopupMenu Detection\_Overlaps,mode=1,popvalue="Ignore",value= #"\"Ignore;Separate;Split\""

PopupMenu Detection\_Overlaps help={"Spikes overlap if the duration between the end of the 1st and the beginning of the 2nd spike is less then 2\*t1/2."}

GroupBox Results\_Peak\_Box, pos={18,27},size={115,231},fColor=(26112,0,10240), title="Spike Param",disable= (tab!=2)

CheckBox Results\_show1,pos={25,45},size={150,14},title="T(max), (s)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show1 help={"Time (seconds) at spike Maximum."}

CheckBox Results\_show2,pos={25,64},size={150,14},title="T(base), (ms)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show2 help={"Duration from the Start to the End of a spike."}

CheckBox Results\_show4,pos={25,83},size={150,14},title="T(1/2), (ms)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show4 help={"Spike width (ms) at 50% of its height."}

CheckBox Results\_show5,pos={25,102},size={150,14},title="I(max), (pA)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show5 help={"Spike height (pA) from its maximum to the background under the maximum."}

CheckBox Results\_show6,pos={25,120},size={150,14},title="Q (pC)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show6 help={"Spike charge (pA/s)."}

CheckBox Results\_show7,pos={25,140},size={150,14},title="Q (molecules)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show7 help={"Peak charge (pA/s) multiplied by 3.121\*10^6."}

CheckBox Results\_show8,pos={25,159},size={150,14},title="T(rise), (ms)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show8 help={"Time between 2 points on spike rising phase. Foot is excluded!"}

SetVariable Results\_Rise\_Low,pos={26,177},size={36,20},title=" ",limits={0,50,5},proc=Change\_Table\_RiseTime,value=root:Quanta:Rise\_Low\_Prc,frame=1,disable= (tab!=2)

SetVariable Results\_Rise\_Low, help={"Lower point (excluding the foot). Calculated using rise phase linear extrapolation. 0-50% of Imax."}

SetVariable Results\_Rise\_Hi,pos={65,177},size={48,20},title="to",limits={55,100,5},proc=Change\_Table\_RiseTime,value=root:Quanta:Rise\_Hi\_Prc,frame=1,disable= (tab!=2)

SetVariable Results\_Rise\_Hi, help={"Upper point : 55-100% of Imax."}

CheckBox Results\_show9,pos={25,198},size={150,14},title="Rise (pA/ms)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show9 help={"Linear slope of spike rising phase. The middle of the slope is found at dI/dt maximum"}

CheckBox Results\_show10,pos={25,217},size={150,14},title="Fall time (ms)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show10 help={"Time between 75 and 25% of the Imax on peak falling phase."}

CheckBox Results\_show11,pos={25,236},size={150,14},title="Fall (...ms)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show11 help={"Linear slope (pA/ms) between 75 and 25% of the Imax or tau (ms) of the exp fit between 75% and the End of the peak."}

GroupBox Results\_Foot\_Box, pos={138,27},size={115,98},fColor=(26112,0,10240), title="Foot Param",disable= (tab!=2)

CheckBox Results\_show12,pos={148,45},size={150,14},title="I(foot), (pA)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show12 help={"Foot current"}

CheckBox Results\_show13,pos={148,64},size={150,14},title="T(foot), (ms)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show13 help={"Foot duration. Time from the spike start to foot end."}

CheckBox Results\_show14,pos={148,83},size={150,14},title="Q(foot), (pC)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show14 help={"Foot area (pC)."}

CheckBox Results\_show15,pos={148,102},size={150,14},title="Q(foot), (molec)",value= 0,proc=Change\_Table,disable= (tab!=2)

CheckBox Results\_show15 help={"Foot area multiplied by 3.121\*10^6 (molecules/pC)."}

GroupBox Results\_FallFit\_Box, pos={138,130},size={115,70},fColor=(26112,0,10240), title="Spike fall",disable= (tab!=2)

PopupMenu Results\_Fallfit,pos={150,150},size={103,21},title="Fit",disable= (tab!=2)

PopupMenu Results\_Fallfit,mode=3,popvalue="DblExp",value= #"\"Line;Exp;DblExp\"",proc=Fall\_fit\_PopMenu

PopupMenu Results\_Fallfit, help={"Choose the function to fit spike falling phase."}

SetVariable Results\_Fall\_Chi, pos={150,177},size={86,16},title="Chi2 ratio",limits={0,1e+03,1},value= Fall\_ChiRatio\_Cutoff,disable=1

SetVariable Results\_Fall\_Chi, help={"DblExp fit will be used only if Chi2s ratio from Exp to DblExp fits > that this threshold. Set to zero to force DblExp!"}

GroupBox Stats\_Stat\_Box, pos={18,50},size={235,55},fColor=(26112,0,10240), title="Statistics",disable= (tab!=3)

PopupMenu Stats\_Population,pos={28,74},size={103,21},title="Stats are calculated using the",disable= (tab!=3)

PopupMenu Stats\_Population,mode=1,popvalue="Median",value= #"\"Median;Mean\"",proc=Stats\_PopMenu

PopupMenu Stats\_Population, help={"Choose 'mean' for normally distributed data. Use 'Stats' button on the Main window."}

GroupBox Stats\_AvePeak\_Box, pos={18,120},size={235,55},fColor=(26112,0,10240), title="Average Spike",disable= (tab!=3)

PopupMenu Stats\_AvePeak,pos={67,143},size={103,21},title="Line up the spikes by",disable= (tab!=3)

PopupMenu Stats\_AvePeak,mode=1,popvalue="Max",value= #"\"Max;Rise\"",proc=Stats\_PopMenu

PopupMenu Stats\_AvePeak, help={"The point by which the spikes will be lined up during averaging. Use 'Avg peak' button on the Main window."}

Endmacro

Window Main\_window() : Graph

PauseUpdate; Silent 1

String fldrSav= GetDataFolder(1)

SetDataFolder root:Quanta:

Display /W=(3,80,549,394.25) Working\_trace\_copy

SetDataFolder fldrSav

ModifyGraph margin(left)=45,gfSize=8,cbRGB=(48896,65280,48896)

ModifyGraph lblLatPos(left)=-80,lblLatPos(bottom)=150

Cursor/P A Working\_trace\_copy 0;Cursor/P B Working\_trace\_copy 0

ControlBar 70

GroupBox Left\_Axis\_Controls,pos={1,178},size={22,103},labelBack=(48896,65280,48896)

Button Zoom\_V\_in,pos={4,212},size={15,15},proc=Zoom\_Up\_Q,title="+"

Button Zoom\_V\_out,pos={4,235},size={15,15},proc=Zoom\_Down\_Q,title="-"

Button Move\_V\_up,pos={4,183},size={15,15},proc=Move\_Up\_Q,title="A"

Button Move\_V\_down,pos={4,261},size={15,15},proc=Move\_Down\_Q,title="V"

GroupBox Bottom\_Axis\_Controls,pos={381,69},size={102,22},labelBack=(48896,65280,48896)

Button Move\_H\_L,pos={385,73},size={15,15},proc=Move\_Left\_Q,title="<"

Button Zoom\_H\_out,pos={413,73},size={15,15},proc=Zoom\_OutHoriz\_Q,title="-"

Button Zoom\_H\_in,pos={436,73},size={15,15},proc=Zoom\_In\_Horiz\_Q,title="+"

Button Move\_H\_R,pos={464,73},size={15,15},proc=Move\_Right\_Q,title=">"

Variable Min\_Freq=Binomial\_to\_Gaussian\_Calc(32767)

GroupBox Separator1,pos={1,0},size={108,69}

Button Smooth,pos={58,45},size={48,21},proc=SmoothBtn\_Q,title="Smooth"

PopupMenu Smoth\_meth,pos={4,3},size={103,21},proc=Smooth\_method\_Q

PopupMenu Smoth\_meth,mode=3,popvalue="Binomial sm. ",value= #"\"LP Gaussian ;HP Gaussian ;Binomial sm. ;Boxcar sm. ;Sav.-Gol. sm.\""

SetVariable Smooth\_F,pos={5,27},size={100,16},title="Hz"

SetVariable Smooth\_F,limits={Min\_Freq+1,(1/(T\_Delta/1000))/2,50},value= root:Quanta:Smoothing\_Factor

SetVariable Smooth\_F,help={"-3dB Cutoff frequency"}

Button Show\_orig,pos={3,45},size={54,21},proc=Show\_original\_rec,title="Show Orig"

GroupBox Overall\_smooth,pos={0,69},size={109,22},labelBack=(48896,65280,48896)

ValDisplay Display\_Overall\_Filter,pos={3,72},size={100,15}, value=root:Quanta:Overall\_Filter,title="Final fc:",mode=1

ValDisplay Display\_Overall\_Filter,format="%.1W1PHz",fsize=9,labelBack=(48896,65280,48896)

ValDisplay Display\_Overall\_Filter,help={"Overall cutoff frequency of several Gaussian filters in a series. Does not work for boxcar and polynomial smoothings."}

GroupBox Separator201,pos={111,1},size={145,68}

GroupBox Bkg\_HiLt,pos={114,3},size={37,25},labelBack=(39168,0,0),frame=0,disable=0

Button Bkg,pos={117,5},size={32,22},proc=Bkg\_noise\_Set,title="Bkg"

Button Bkg,help={"Position cursor A (round) at the beginning and cursor B (square) at the end of the trace's part, which does not have any spikes."}

Button Find\_Pk,pos={153,5},size={98,22},proc=Peak\_finder,title="Analyse Trace"

SetVariable Set\_threshold,pos={118,30},size={133,16},title="Peak threshold=",help={"A spike is detected if dI/dt>=SD(dI)\*threshold"}

SetVariable Set\_threshold,limits={-Inf,Inf,0.5},value= root:Quanta:Detection\_Mult

SetVariable Foot\_threshold,pos={118,48},size={133,16},title="Foot threshold =",help={"Foot is detected if I(foot) >SD(I)\*threshold"}

SetVariable Foot\_threshold,limits={-Inf,Inf,0.5},value= root:Quanta:Detection\_Foot\_Mult

GroupBox Separator2,pos={258,1},size={147,53}

ValDisplay No\_found,pos={338,6},size={59,15},title="of"

ValDisplay No\_found,limits={0,0,0},barmisc={0,1000}

ValDisplay No\_found,value= #"root:Quanta:Total\_peaks\_number"

SetVariable ID,pos={265,6},size={67,16},proc=Peak\_locator\_ID,title="ID"

SetVariable ID,limits={1,0,1},value= root:Quanta:Peak\_ID

Button See\_all,pos={298,26},size={65,22},proc=Zoom\_Trace\_In\_Out,title="Zoom In"

Button Last,pos={264,26},size={33,22},proc=Peak\_surf,title="<<"

Button Next,pos={364,26},size={33,22},proc=Peak\_surf,title=">>"

GroupBox Separator3,pos={408,1},size={228,46},labelBack=(39168,0,0),frame=0

Button Del\_All,pos={411,3},size={80,20},proc=Delete\_all\_peaks,title="Delete All"

Button Del\_All,help={"Delete all spikes"}

Button Del,pos={411,25},size={80,20},proc=Delete\_peak,title="Del Peak"

Button Del,help={"Currently selected spike will be deleted. If this peak is a part of splitted spikes, it will be added to the peak on its left."}

Button Spllit,pos={492,3},size={80,20},proc=Split,title="Split"

Button Spllit,help={"Use cursor A (round) to mark the split point."}

Button New\_pk,pos={492,25},size={80,20},proc=Add\_Peak\_Manually,title="Add Peak"

Button New\_pk,help={"Cursor A (round) should be at the beginning and cursor B (square) at the end of the spike."}

Button NewBase,pos={574,25},size={60,20},proc=New\_Baseline,title="New Base"

Button NewBase,help={"Cursor A (round) should be at the beginning and cursor B (square) at the end of a spike."}

Button NewMax,pos={574,3},size={60,20},proc=New\_T\_Max,title="New Max"

Button NewMax,help={"Use cursor A (round) to mark a new spike Maximum."}

GroupBox Separator4,pos={638,0},size={86,49},labelBack=(48896,65280,65280)

Button Statistics,pos={641,3},size={80,24},proc=Show\_stats,title="Stats"

Button Statistics,help={"Calculates Means, SD and SE for all parameters"}

Button Avg\_Peak,pos={641,28},size={80,18},proc=Average\_peaks,title="Avg peak"

Button Avg\_Peak,help={"Averages the detected spikes"}

Slider X\_Slider,pos={258,55},size={349,13},proc=Slider\_Horiz\_Q

Slider X\_Slider,limits={0,1,0},value= 0.654716981132076,side= 0,vert= 0,ticks= 0

Button Reset\_Slider\_Btn,pos={610,50},size={25,18},proc=Slider\_Reset,title="res"

Button Reset\_Slider\_Btn,help={"If slider does not work push here!"}

Button Save\_zoom,pos={641,50},size={80,18},proc=Save\_zoomed\_trace,title="Save Zoom"

Button Save\_zoom,help={"Saves recording in the Main window. Don't forget to include some area containing noise for future analysis."}

EndMacro

//\_\_\_Zoom window controls\_\_\_\_

Window Zoom\_Win() : Graph

PauseUpdate; Silent 1

SetDataFolder root:Quanta:

Variable Resolution=ScreenResolution

Variable Zoom\_Win\_width=207\*96/Resolution

Display/K=1 /W=(555,80,555+Zoom\_Win\_width,299.75) root:Quanta:Zoomed\_peak,root:Quanta:Fall\_phase,root:Quanta:Rise\_phase

ModifyGraph rgb(Zoomed\_peak)=(0,0,0),rgb(Fall\_phase)=(65280,0,0),rgb(Rise\_phase)=(65280,0,0)

ModifyGraph gfSize=7, axOffset(left)=-3, lblLatPos(left)=-90, lblRot(left)=-90

Label left "\\f01\\Z07pA"

Label bottom "\\f01\\Z07s"

SetAxis/A

Cursor/P A Zoomed\_peak 0;Cursor/P B Zoomed\_peak 0

ControlBar 41

GroupBox Separator2,pos={1,0},size={77,40}

SetVariable ID,pos={6,3},size={67,16},proc=Peak\_locator\_ID,title="ID"

SetVariable ID,limits={1,0,1},value= root:Quanta:Peak\_ID

Button Last,pos={6,20},size={33,18},proc=Peak\_surf,title="<<"

Button Next,pos={41,20},size={33,18},proc=Peak\_surf,title=">>"

CheckBox Zoom\_to\_Foot,pos={80,0},size={39,14},proc=Zoom\_to\_foot\_proc,title="Foot"

CheckBox Zoom\_to\_Rise,pos={142,0},size={39,14},proc=Zoom\_to\_foot\_proc,title="Rise"

CheckBox Zoom\_to\_Fall,pos={204,0},size={34,14},proc=Zoom\_to\_foot\_proc,title="Fall"

GroupBox Separ\_WholePeak,pos={79,14},size={189,25},labelBack=(39168,0,0),frame=0

Button Del,pos={82,17},size={60,20},proc=Delete\_peak,title="Del Peak", disable=0

Button Del,help={"Currently selected spike will be deleted. If this peak is a part of splitted spikes, it will be added to the peak on its left."}

Button NewBase,pos={144,17},size={60,20},proc=New\_Baseline,title="New Base", disable=0

Button NewBase,help={"Cursor A (round) should be at the beginning and cursor B (square) at the end of a Peak."}

Button NewMax,pos={206,17},size={60,20},proc=New\_T\_Max,title="New Max", disable=0

Button NewMax,help={"Use cursor A (round) to mark a new spike Maximum."}

GroupBox Separator3,pos={79,41},size={189,25},labelBack=(39168,0,0),frame=0,disable=1

Button FootDelete,pos={82,44},size={60,18},proc=Delete\_Foot,title="Del Foot",disable=1

Button FootDelete,help={"Will delete the Foot of currently selected Peak."}

Button FootNew\_H,pos={144,44},size={60,18},proc=New\_Foot\_I,title="Foot H",disable=1

Button FootNew\_H,help={"Use cursor A (round) to set new Foot current."}

Button New\_Rise,pos={144,44},size={60,18},proc=New\_Rise,title="New Rise",disable=1

Button New\_Rise,help={"Cursor A (round) should be on the lower and cursor B (square) on the higher part of peak's rise."}

PopupMenu Fall\_fit\_change,pos={173,43},size={103,21},title="Fit",disable=1

PopupMenu Fall\_fit\_change,mode=3,popvalue=root:Quanta:Fit\_method,value= #"\"Line;Exp;DblExp\"",proc=Fall\_fit\_PopMenu\_Single

PopupMenu Fall\_fit\_change, help={"Choose the function to fit the falling phase of the selected spike"}

PopupMenu Fall\_Extrap\_change,pos={131,78},size={103,21},title="Extrapolate",disable=1

PopupMenu Fall\_Extrap\_change,mode=2,popvalue=root:Quanta:Fit\_method,value= #"\"Line;Exp\"",proc=Fall\_Extrap\_PopMenu\_Single

PopupMenu Fall\_Extrap\_change, help={"Choose the function to extrapolate the falling phase of selected peak."}

SetVariable Extrap\_Tau\_Set title="Extrap Tau",pos={133,104},size={110,16},proc=Extrap\_Tau\_SetVar, disable=1

SetVariable Extrap\_Tau\_Set value=Fall\_Tau\_Extrap, format="%.2f",labelBack=(65535,65535,65535)

EndMacro

Function Extrap\_Tau\_SetVar(ctrlName,varNum,varStr,varName) : SetVariableControl

String ctrlName

Variable varNum

String varStr

String varName

SetDataFolder root:Quanta:

Wave/T Fall\_fit\_Extrap=Fall\_fit\_Extrap

NVAR Peak\_ID=Peak\_ID

Fall\_fit\_Extrap[Peak\_ID-1]=ReplaceStringByKey("Tau", Fall\_fit\_Extrap[Peak\_ID-1], varStr)

String First\_and\_Last=Check\_for\_Separated\_peaks(Peak\_ID-1)

Variable First\_separated\_pnt=str2num(StringFromList(0,First\_and\_Last))

Variable Last\_separated\_pnt=str2num(StringFromList(1,First\_and\_Last))

Calc\_Separated\_peak\_param(First\_separated\_pnt,Last\_separated\_pnt)

Draw\_Lines\_All(Peak\_ID-1)

End

Function Zoom\_to\_Foot\_Proc (theTag,checked) : CheckBoxControl

String theTag

Variable checked

NVAR Peak\_ID=Peak\_ID

NVAR Total\_peaks\_number=Total\_peaks\_number

strswitch (theTag)

case "Zoom\_to\_Foot":

CheckBox Zoom\_to\_Rise value=0

CheckBox Zoom\_to\_Fall value=0

break

case "Zoom\_to\_Rise":

CheckBox Zoom\_to\_Foot value=0

CheckBox Zoom\_to\_Fall value=0

break

case "Zoom\_to\_Fall":

CheckBox Zoom\_to\_Foot value=0

CheckBox Zoom\_to\_Rise value=0

break

endswitch

If(Total\_peaks\_number==0)

abort

endif

Draw\_lines\_zoom\_window(Peak\_ID-1)

End

Function Change\_Table (ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

SetDataFolder $"root:Quanta"

NVAR Show\_Time=Show\_Time

NVAR Show\_Base=Show\_Base

NVAR Show\_Width=Show\_Width

NVAR Show\_H=Show\_H

NVAR Show\_Q=Show\_Q

NVAR Show\_Molec=Show\_Molec

NVAR Show\_Rise\_t=Show\_Rise\_t

NVAR Show\_Rise\_r=Show\_Rise\_r

NVAR Show\_Fall\_t=Show\_Fall\_t

NVAR Show\_Fall\_r=Show\_Fall\_r

NVAR Show\_Ft\_H=Show\_Ft\_H

NVAR Show\_Ft\_width=Show\_Ft\_width

NVAR Show\_Ft\_Q=Show\_Ft\_Q

NVAR Show\_Ft\_molec=Show\_Ft\_molec

NVAR Rise\_Low\_Prc=Rise\_Low\_Prc

NVAR Rise\_Hi\_Prc=Rise\_Hi\_Prc

SVAR Values\_to\_show=Values\_to\_show

Values\_to\_show="Peak\_Num"

SVAR Stats\_names=Stats\_names

Stats\_names="\tIntSp[ms]\tID#\t"

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_show1; Show\_Time=V\_Value

endif

If(Show\_Time==1)

Values\_to\_show+=",T\_Max"

Stats\_names+="Max[s]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show2; Show\_Base=V\_Value

endif

If(Show\_Base==1)

Values\_to\_show+=",Peak\_Base"

Stats\_names+="Base[ms]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show4; Show\_Width=V\_Value

endif

If(Show\_Width==1)

Values\_to\_show+=",Peak\_t05"

Stats\_names+="t1/2[ms]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show5; Show\_H=V\_Value

endif

If(Show\_H==1)

Values\_to\_show+=",Peak\_Imax"

Stats\_names+="Imax [pA]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show6; Show\_Q=V\_Value

endif

If(Show\_Q==1)

Values\_to\_show+=",Peak\_Q"

Stats\_names+="Q [pC]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show7; Show\_Molec=V\_Value

endif

If(Show\_Molec==1)

Values\_to\_show+=",Peak\_Molec"

Stats\_names+="Molec\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show8; Show\_Rise\_t=V\_Value

endif

If(Show\_Rise\_t==1)

Values\_to\_show+=",Rise\_time"

Stats\_names+="Rise("+num2str(Rise\_Low\_Prc)+"-"+num2str(Rise\_Hi\_Prc)+")[ms]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show9; Show\_Rise\_r=V\_Value

endif

If(Show\_Rise\_r==1)

Values\_to\_show+=",Rise\_slope"

Stats\_names+="Rise[pA/ms]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show10; Show\_Fall\_t=V\_Value

endif

If(Show\_Fall\_t==1)

Values\_to\_show+=",Fall\_time"

Stats\_names+="Fall(75-25) [ms]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show11; Show\_Fall\_r=V\_Value

endif

If(Show\_Fall\_r==1)

Values\_to\_show+=",Fall\_fit"

Stats\_names+="Fall Fit\t"

SVAR Fit\_method=Fit\_method

If (cmpstr(Fit\_method,"Line")==0)

Values\_to\_show+=",Fall\_slope"

Stats\_names+="Fall slope[pA/ms]\t"

else

Values\_to\_show+=",Fall\_slope,Fall\_slope2"

Stats\_names+="Fall tau1[ms]\tFall tau2[ms]\t"

endif

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show12; Show\_Ft\_H=V\_Value

endif

If(Show\_Ft\_H==1)

Values\_to\_show+=",Foot\_I"

Stats\_names+="I(foot) [pA]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show13; Show\_Ft\_width=V\_Value

endif

If(Show\_Ft\_width==1)

Values\_to\_show+=",Foot\_W"

Stats\_names+="T(foot) [ms]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show14; Show\_Ft\_Q=V\_Value

endif

If(Show\_Ft\_Q==1)

Values\_to\_show+=",Foot\_Q"

Stats\_names+="Q(foot) [pC]\t"

endif

if(CheckName("Options\_Tab\_Panels", 9)!=0)

ControlInfo/W=Options\_Tab\_Panels Results\_Show15; Show\_Ft\_molec=V\_Value

endif

If(Show\_Ft\_molec==1)

Values\_to\_show+=",Foot\_Molec"

Stats\_names+="Q(Foot) [Molec]"

endif

Stats\_names+="\r\r"

Dowindow/K Peak\_stats\_Table1

Peak\_stats\_Table()

AutoPositionWindow/E/M=0/R=Main\_window Peak\_stats\_Table1

Dowindow/F Options\_Tab\_Panels

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Peak\_ID=Peak\_ID

NVAR Show\_Legend=Show\_Legend

If((Total\_peaks\_number)&(Show\_Legend==1))

Generate\_annotation(Peak\_ID-1)

endif

End

Function Change\_Table\_RiseTime(ctrlName,varNum,varStr,varName) : SetVariableControl

String ctrlName

Variable varNum

String varStr

String varName

Change\_Table ("q",1)

End

Function Peak\_stats\_Table()

SetDataFolder $"root:Quanta"

SVAR Values\_to\_show=Values\_to\_show

NVAR Scrn\_hight\_Points=Scrn\_hight\_Points

Variable Resolution=ScreenResolution

Variable Table\_Top=Scrn\_hight\_Points-180\*96/Resolution

Variable Table\_Bottom=Scrn\_hight\_Points-75\*96/Resolution

String Things\_to\_go="Edit/K=1/W=(2,"+num2str(Table\_Top)+",50,"+num2str(Table\_Bottom)+") "+ Values\_to\_show

// String Things\_to\_go="Edit/K=1/W=(2,415,50,415+"+num2str(Table\_hight)+") "+ Values\_to\_show

execute Things\_to\_go

DoWindow/C Peak\_stats\_Table1

DoWindow/T Peak\_stats\_Table1,"Spikes statistics"

execute "Peak\_stats\_Table\_formats()"

NVAR Table\_L=Table\_L

MoveWindow 2,Table\_Top,Table\_L,Table\_Bottom

KillVariables /Z Table\_L

End

Window Peak\_stats\_Table\_formats()

SetDataFolder $"root:Quanta"

Variable/G Table\_L=40

ModifyTable size(Point)=8,width(Point)=0,size(Peak\_Num)=8,style(Peak\_Num)=1,width(Peak\_Num)=20, title(Peak\_Num)="Id#",rgb(Peak\_Num)=(65280,0,0),font(Peak\_Num)="Arial"

if(strsearch(Values\_to\_show,"T\_Max",0)!=-1)

ModifyTable size(T\_Max)=7,font(T\_Max)="Arial",style(T\_Max)=1,format(T\_Max)=3,digits(T\_Max)=3,width(T\_Max)=35,title(T\_Max)="Max [s]",trailingZeros(T\_Max)=1

Table\_L+=35

endif

if(strsearch(Values\_to\_show,"Peak\_Base",0)!=-1)

ModifyTable size(Peak\_Base)=8,font(Peak\_Base)="Arial",style(Peak\_Base)=1,format(Peak\_Base)=3,digits(Peak\_Base)=3,width(Peak\_Base)=40, title(Peak\_Base)="Base [ms]",trailingZeros(Peak\_Base)=1,rgb(Peak\_Base)=(0,12800,52224)

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Peak\_t05",0)!=-1)

ModifyTable size(Peak\_t05)=8,font(Peak\_t05)="Arial",style(Peak\_t05)=1,format(Peak\_t05)=3,digits(Peak\_t05)=3,width(Peak\_t05)=40, title(Peak\_t05)="t1/2 [ms]",trailingZeros(Peak\_t05)=1,rgb(Peak\_t05)=(0,12800,52224)

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Peak\_Imax",0)!=-1)

ModifyTable style(Peak\_Imax)=1,font(Peak\_Imax)="Arial",digits(Peak\_Imax)=2,width(Peak\_Imax)=40,title(Peak\_Imax)="Imax [pA]",size(Peak\_Imax)=8,rgb(Peak\_Imax)=(0,12800,52224),trailingZeros(Peak\_Imax)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Peak\_Q",0)!=-1)

ModifyTable size(Peak\_Q)=8,font(Peak\_Q)="Arial",style(Peak\_Q)=1,width(Peak\_Q)=40,title(Peak\_Q)="Q [pC]",rgb(Peak\_Q)=(0,12800,52224),trailingZeros(Peak\_Q)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Peak\_Molec",0)!=-1)

ModifyTable size(Peak\_Molec)=7,font(Peak\_Molec)="Arial",style(Peak\_Molec)=1,sigDigits(Peak\_Molec)=4,width(Peak\_Molec)=40,title(Peak\_Molec)="Molecules",rgb(Peak\_Molec)=(0,12800,52224),trailingZeros(Peak\_Molec)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Rise\_time",0)!=-1)

Variable/G Rise\_Low\_Prc,Rise\_Hi\_Prc

String Column\_name="Rise("+num2str(Rise\_Low\_Prc)+"-"+num2str(Rise\_Hi\_Prc)+")[ms]\t"

ModifyTable size(Rise\_time)=8,font(Rise\_time)="Arial", style(Rise\_time)=1,sigDigits(Rise\_time)=3,width(Rise\_time)=40,title(Rise\_time)=Column\_name,rgb(Rise\_time)=(0,12800,52224),trailingZeros(Rise\_time)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Rise\_slope",0)!=-1)

ModifyTable size(Rise\_slope)=8,font(Rise\_slope)="Arial",style(Rise\_slope)=1,sigDigits(Rise\_slope)=3,width(Rise\_slope)=40,title(Rise\_slope)="Rise[pA/ms]",rgb(Rise\_slope)=(0,12800,52224),trailingZeros(Rise\_slope)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Fall\_time",0)!=-1)

ModifyTable sigDigits(Fall\_time)=3,font(Fall\_time)="Arial",width(Fall\_time)=40,title(Fall\_time)="Fall(75-25)[ms]",size(Fall\_time)=8,style(Fall\_time)=1,rgb(Fall\_time)=(0,12800,52224),trailingZeros(Fall\_time)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Fall\_fit",0)!=-1)

ModifyTable size(Fall\_fit)=7,font(Fall\_fit)="Arial",style(Fall\_fit)=1,format(Fall\_fit)=3,digits(Fall\_fit)=3,width(Fall\_fit)=35,title(Fall\_fit)="Fall fit"

Table\_L+=35

endif

if(strsearch(Values\_to\_show,"Fall\_slope",0)!=-1)

String/G Fit\_method

ModifyTable size(Fall\_slope)=8,font(Fall\_slope)="Arial",style(Fall\_slope)=1,sigDigits(Fall\_slope)=3,width(Fall\_slope)=40,rgb(Fall\_slope)=(0,12800,52224),trailingZeros(Fall\_slope)=1

If (cmpstr(Fit\_method,"Line")==0)

ModifyTable title(Fall\_slope)="Slope[pA/ms]"

else

ModifyTable title(Fall\_slope)="Tau1 [ms]"

endif

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Fall\_slope2",0)!=-1)

ModifyTable size(Fall\_slope2)=8,font(Fall\_slope2)="Arial",style(Fall\_slope2)=1,sigDigits(Fall\_slope2)=3,width(Fall\_slope2)=40,title(Fall\_slope2)="Tau2 [ms]",rgb(Fall\_slope2)=(0,12800,52224),trailingZeros(Fall\_slope2)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Foot\_I",0)!=-1)

ModifyTable width(Foot\_I)=40,font(Foot\_I)="Arial",title(Foot\_I)="I(foot) [pA]",rgb(Foot\_I)=(26112,0,0),size(Foot\_I)=8,style(Foot\_I)=1,digits(Foot\_I)=2,trailingZeros(Foot\_I)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Foot\_W",0)!=-1)

ModifyTable width(Foot\_W)=40,font(Foot\_W)="Arial",title(Foot\_W)="T(foot) [ms]",rgb(Foot\_W)=(26112,0,0),size(Foot\_W)=8,style(Foot\_W)=1,digits(Foot\_W)=2,trailingZeros(Foot\_W)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Foot\_Q",0)!=-1)

ModifyTable size(Foot\_Q)=8,font(Foot\_Q)="Arial",style(Foot\_Q)=1,width(Foot\_Q)=40,title(Foot\_Q)="Q(foot) [pC]",rgb(Foot\_Q)=(26112,0,0),trailingZeros(Foot\_Q)=1

Table\_L+=40

endif

if(strsearch(Values\_to\_show,"Foot\_Molec",0)!=-1)

ModifyTable size(Foot\_Molec)=8,font(Foot\_Molec)="Arial",style(Foot\_Molec)=1,sigDigits(Foot\_Molec)=4,width(Foot\_Molec)=40,title(Foot\_Molec)="Q(Foot) [Molec]",rgb(Foot\_Molec)=(26112,0,0),trailingZeros(Foot\_Molec)=1

Table\_L+=40

endif

// MoveWindow 2,415,Table\_L,510

EndMacro

Function Recalculate\_All\_Peaks(Do\_what)

String Do\_what

SetDataFolder $"root:Quanta"

Wave Working\_trace\_copy=Working\_trace\_copy

Wave T\_Max=T\_Max

Wave T\_Bkg1=T\_Bkg1

Wave T\_Bkg2=T\_Bkg2

Wave Foot\_end=Foot\_end

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Peak\_ID=Peak\_ID

NVAR Bkg\_noise\_Start=Bkg\_noise\_Start

NVAR Bkg\_noise\_End=Bkg\_noise\_End

Variable/G Bkg\_noise\_Start\_p

Variable/G Bkg\_noise\_End\_p

If(cmpstr(Do\_what,"Save\_old")==0)

Duplicate/O T\_Max T\_Max\_pnt

Duplicate/O T\_Bkg1 T\_Bkg1\_pnt

Duplicate/O T\_Bkg2 T\_Bkg2\_pnt

T\_Max\_pnt[0,]=x2pnt(Working\_trace\_copy,T\_Max[p])

T\_Bkg1\_pnt[0,]=x2pnt(Working\_trace\_copy,T\_Bkg1[p])

T\_Bkg2\_pnt[0,]=x2pnt(Working\_trace\_copy,T\_Bkg2[p])

Bkg\_noise\_Start\_p=x2pnt(Working\_trace\_copy,Bkg\_noise\_Start)

Bkg\_noise\_End\_p=x2pnt(Working\_trace\_copy,Bkg\_noise\_End)

endif

If(cmpstr(Do\_what,"Recalc\_new")==0)

Variable i, Max\_X, Start\_X, End\_X

Bkg\_noise\_Start=pnt2x(Working\_trace\_copy,Bkg\_noise\_Start\_p)

Bkg\_noise\_End=pnt2x(Working\_trace\_copy,Bkg\_noise\_End\_p)

Bkg\_noise\_Calc()

T\_Max[0,]=pnt2x(Working\_trace\_copy,T\_Max\_pnt[p])

T\_Bkg1[0,]=pnt2x(Working\_trace\_copy,T\_Bkg1\_pnt[p])

T\_Bkg2[0,]=pnt2x(Working\_trace\_copy,T\_Bkg2\_pnt[p])

for(i=0;i<Total\_peaks\_number;i+=1)

Calc\_Peak\_Parameters(i,T\_Bkg1[i],T\_Bkg2[i])

endfor

KillVariables /Z Bkg\_noise\_Start\_p,Bkg\_noise\_End\_p

Killwaves/Z T\_Max\_pnt,T\_Bkg1\_pnt,T\_Bkg2\_pnt

Peak\_ID=1

Draw\_lines\_All(0)

endif

End

Function Recalculate\_Peaks\_Btn(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

Wave Rise\_Midpoint=Rise\_Midpoint

Wave Rise\_time=Rise\_time

Wave/T Fall\_fit=Fall\_fit

NVAR Total\_peaks\_number=Total\_peaks\_number

Fall\_fit=""

Rise\_Midpoint=0

Rise\_time=0

If (Total\_peaks\_number)

Recalculate\_All\_Peaks("Save\_old")

Recalculate\_All\_Peaks("Recalc\_new")

endif

Dowindow/F Options\_Tab\_Panels

End

Function Adjust\_Trace\_Scales(ctrlName,varNum,varStr,varName) : SetVariableControl

String ctrlName

Variable varNum

String varStr

String varName

WAVE Working\_trace\_copy=Working\_trace\_copy

NVAR T\_Start=T\_Start

NVAR T\_End=T\_End

NVAR T\_Delta=T\_Delta

NVAR Gain=Gain

NVAR Gain\_Temp=Gain\_Temp

Variable N\_pnts=numpnts(Working\_trace\_copy)

strswitch(ctrlName)

case "Scales\_Start\_time":

ControlInfo Scales\_Change\_End

If(V\_Value)

T\_Delta=((T\_End-T\_Start)/N\_pnts)\*1000

else

T\_End=T\_Start+T\_delta/1000\*N\_pnts

endif

break

case "Scales\_End\_time":

T\_Delta=((T\_End-T\_Start)/N\_pnts)\*1000

break

case "Scales\_Delta\_time":

T\_End=T\_Start+T\_delta/1000\*N\_pnts

break

endswitch

Button Recalculate\_btn fColor=(0,0,0),win=Options\_Tab\_Panels

Variable Old\_T\_Delta=(pnt2x(Working\_trace\_copy,1)-pnt2x(Working\_trace\_copy,0))\*1000

If(Old\_T\_Delta!=T\_Delta)

Button Recalculate\_btn fColor=(0,39168,0),win=Options\_Tab\_Panels

endif

If((T\_Start!=pnt2x(Working\_trace\_copy,0))%|(T\_End!=pnt2x(Working\_trace\_copy,(numpnts(Working\_trace\_copy)-1))))

Button Recalculate\_btn fColor=(0,39168,0),win=Options\_Tab\_Panels

endif

If(Gain\_Temp!=Gain)

Button Recalculate\_btn fColor=(0,39168,0),win=Options\_Tab\_Panels

endif

End

Function Change\_trace\_scales(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

WAVE Working\_trace\_copy=Working\_trace\_copy

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Gain=Gain

NVAR Gain\_Temp=Gain\_Temp

NVAR Bkg\_noise\_I=Bkg\_noise\_I

NVAR Bkg\_noise\_dI=Bkg\_noise\_dI

NVAR Bkg\_noise\_Start=Bkg\_noise\_Start

NVAR Bkg\_noise\_End=Bkg\_noise\_End

NVAR T\_Start=T\_Start

NVAR T\_End=T\_End

NVAR T\_Delta=T\_Delta

Variable Changed=0

If(Gain\_Temp!=Gain)

if(Total\_peaks\_number)

String Massage="Do You want to recalculate the parameters of the existing spikes? \r Press 'Yes' to Recalculate.\r Press 'No' to Delete the spikes.\rNote: Will not work on Separated and Split spikes!"

DoAlert 2, Massage

switch(V\_Flag)

case 1:

Recalculate\_All\_Peaks("Save\_old")

Change\_trace\_Y\_scale()

Recalculate\_All\_Peaks("Recalc\_new")

break

case 2:

De\_novo()

GroupBox Bkg\_HiLt, win=Main\_window,disable=0

Bkg\_noise\_I=0

Bkg\_noise\_dI=0

Bkg\_noise\_Start=0

Bkg\_noise\_End=0

Change\_trace\_Y\_scale()

break

case 3:

abort

break

endswitch

else

Change\_trace\_Y\_scale()

endif

endif

Variable Old\_T\_Delta=(pnt2x(Working\_trace\_copy,1)-pnt2x(Working\_trace\_copy,0))\*1000

If(Old\_T\_Delta!=T\_Delta)

Change\_trace\_X\_scale()

endif

If((T\_Start!=pnt2x(Working\_trace\_copy,0))%|(T\_End!=pnt2x(Working\_trace\_copy,(numpnts(Working\_trace\_copy)-1))))

Change\_trace\_X\_scale()

endif

Button Recalculate\_btn fColor=(0,0,0),win=Options\_Tab\_Panels

End

Function Change\_trace\_X\_scale()

SetDataFolder $"root:Quanta"

NVAR T\_Start=T\_Start

NVAR T\_End=T\_End

NVAR Peak\_ID=Peak\_ID

NVAR Bkg\_noise\_I=Bkg\_noise\_I

NVAR Bkg\_noise\_dI=Bkg\_noise\_dI

NVAR Bkg\_noise\_Start=Bkg\_noise\_Start

NVAR Bkg\_noise\_End=Bkg\_noise\_End

NVAR Total\_peaks\_number=Total\_peaks\_number

WAVE Working\_trace\_copy=Working\_trace\_copy

WAVE Orig\_trace\_copy=Orig\_trace\_copy

NVAR T\_Delta=T\_Delta

NVAR Gain=Gain

NVAR Gain\_Temp=Gain\_Temp

Variable N\_pnts=numpnts(Working\_trace\_copy)

if(Total\_peaks\_number)

String Massage="All spikes will be Deleted! \r Press 'Yes' to proceed or 'No' to to Cancel"

DoAlert 1, Massage

If(V\_Flag==1)

De\_novo()

else

abort

endif

endif

Bkg\_noise\_I=0

Bkg\_noise\_dI=0

Bkg\_noise\_Start=0

Bkg\_noise\_End=0

GroupBox Bkg\_HiLt, win=Main\_window,disable=0

Dowindow/F Main\_window

GetAxis /Q bottom

Variable P\_min=x2pnt(Working\_trace\_copy,V\_min)

Variable P\_max=x2pnt(Working\_trace\_copy,V\_max)

ControlInfo/W=Options\_Tab\_Panels Scales\_Change\_End

If(V\_Value)

SetScale/I x T\_Start,T\_End,"s", Working\_trace\_copy

SetScale/I x T\_Start,T\_End,"s", Orig\_trace\_copy

T\_Delta=((T\_End-T\_Start)/N\_pnts)\*1000

else

SetScale/P x T\_Start,T\_Delta/1000,"s", Working\_trace\_copy

SetScale/P x T\_Start,T\_Delta/1000,"s", Orig\_trace\_copy

T\_End=T\_Start+T\_delta/1000\*N\_pnts

endif

Filter\_Limits()

SetAxis bottom pnt2x(Working\_trace\_copy,P\_min),pnt2x(Working\_trace\_copy,P\_max)

Dowindow/F Options\_Tab\_Panels

End

Function Change\_trace\_Y\_scale()

SetDataFolder $"root:Quanta"

NVAR Total\_peaks\_number=Total\_peaks\_number

NVAR Gain=Gain

NVAR Gain\_Temp=Gain\_Temp

Wave Working\_trace\_copy=Working\_trace\_copy

Wave Orig\_trace\_copy=Orig\_trace\_copy

Working\_trace\_copy\*=Gain\_Temp/Gain

Orig\_trace\_copy\*=Gain\_Temp/Gain

SetAxis/A left

Gain=Gain\_Temp

Dowindow/F Options\_Tab\_Panels

End

Function Show\_original\_rec(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

If(exists("Orig\_trace\_copy")!=1)

abort

endif

If(cmpstr(ctrlName, "Show\_orig")==0)

AppendToGraph Orig\_trace\_copy

ModifyGraph rgb(Orig\_trace\_copy)=(52224,52224,52224)

ReorderTraces Working\_trace\_copy,{Orig\_trace\_copy}

Button Show\_orig rename=Hide\_orig, title="Hide Orig"

else

RemoveFromGraph/Z Orig\_trace\_copy

Button Hide\_orig rename=Show\_orig, title="Show Orig"

endif

End

//\_\_\_\_\_\_\_\_Digital Filters\_\_\_\_\_\_\_\_\_\_\_

Function SmoothBtn\_Q(ctrlname) : ButtonControl

String ctrlname

SetDataFolder $"root:Quanta"

If(exists("Orig\_trace\_copy")!=1)

abort

endif

Wave diff1=diff1

NVAR Smoothing\_Factor=Smoothing\_Factor

NVAR Overall\_Filter=Overall\_Filter

String Traces\_Names=TraceNameList("","",1)

String One\_trace

String TheTrace\_Name=StringFromList(0,Traces\_Names)

Variable q=0

Do

One\_trace=Stringfromlist(q,Traces\_Names)

If(strlen(One\_trace)==0)

break

endif

If(stringmatch(One\_trace, "Working\_Trace\_copy")==1)

TheTrace\_Name=One\_trace

break

endif

q+=1

while(1)

ControlInfo Smoth\_meth

if(V\_value==1)

// Low-Pass Gaussian filter

Gaussian\_Filter(TheTrace\_Name,"Low")

If(Overall\_Filter)

Overall\_Filter=SQRT(((Overall\_Filter\*Smoothing\_Factor)^2)/(Overall\_Filter^2+Smoothing\_Factor^2))

else

Overall\_Filter=Smoothing\_Factor

endif

endif

if(V\_value==2)

// High-Pass Gaussian filter

Gaussian\_Filter(TheTrace\_Name,"High")

endif

if(V\_value==3)

// Binomial (Gaussian) smoothing

Variable Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Smoothing\_Factor)

Variable Real\_cutoff=Binomial\_to\_Gaussian\_Calc(Binomial\_coeff)

Print "Trace was filtered with "+num2str(Real\_cutoff)+"Hz (Binomial "+num2str(Binomial\_coeff)+") -3dB Gaussian filter."

Smooth Binomial\_coeff, $TheTrace\_Name

If(Overall\_Filter)

Overall\_Filter=SQRT(((Overall\_Filter\*Smoothing\_Factor)^2)/(Overall\_Filter^2+Smoothing\_Factor^2))

else

Overall\_Filter=Smoothing\_Factor

endif

endif

if(V\_value==4)

// Boxcar (sliding average) smoothing

Smooth/B Smoothing\_Factor,$TheTrace\_Name

endif

if(V\_value==5)

// Savitzky-Golay (polynomial) smoothing

Smooth/S=2 Smoothing\_Factor, $TheTrace\_Name

endif

// Wave Working\_Trace\_copy=Working\_Trace\_copy

// Duplicate/O Orig\_trace\_copy PostFilter\_Residuals

// PostFilter\_Residuals-=Working\_Trace\_copy

SetDrawLayer /K UserFront

End

Function Smooth\_method\_Q(ctrlName,popNum,popStr) : PopupMenuControl

String ctrlName

Variable popNum

String popStr

SetDataFolder $"root:Quanta"

NVAR Smoothing\_Factor=Smoothing\_Factor

NVAR T\_Delta=T\_Delta

Filter\_Limits()

if(popNum==1)

Smoothing\_Factor=500

endif

if(popNum==2)

Smoothing\_Factor=0.1

endif

if(popNum==3)

Variable Min\_Freq=Binomial\_to\_Gaussian\_Calc(32767)

Smoothing\_Factor=max(500,Min\_Freq+1)

endif

if(popNum==4)

Smoothing\_Factor=25

endif

if(popNum==5)

Smoothing\_Factor=5

endif

End

Function Filter\_Limits()

SetDataFolder $"root:Quanta"

NVAR T\_Delta=T\_Delta

ControlInfo/W=Main\_window Smoth\_meth

if(V\_value==1)

SetVariable Smooth\_F,win=Main\_window, title="Hz",limits={1,(1/(T\_Delta/1000))/2,10}

SetVariable Smooth\_F,win=Main\_window, help={"Low-pass cutoff frequency (-3dB) for Gaussian filter."}

endif

if(V\_value==2)

SetVariable Smooth\_F,win=Main\_window, title="Hz",limits={0.0000001,(1/(T\_Delta/1000))/2,0.01}

SetVariable Smooth\_F,win=Main\_window, help={"High-pass cutoff frequency (-3dB) for Gaussian filter."}

endif

if(V\_value==3)

Variable Min\_Freq=Binomial\_to\_Gaussian\_Calc(32767)

SetVariable Smooth\_F,win=Main\_window, title="Hz",limits={Min\_Freq+1,(1/(T\_Delta/1000))/2,50}

SetVariable Smooth\_F,win=Main\_window, help={"Low-pass cutoff frequency (-3dB) for Gaussian filter."}

endif

if(V\_value==4)

SetVariable Smooth\_F,win=Main\_window, title="Points",limits={1,32767,10}

SetVariable Smooth\_F,win=Main\_window, help={"The number of points in the smoothing window for Boxcar smoothing."}

endif

if(V\_value==5)

SetVariable Smooth\_F,win=Main\_window, title="Points",limits={5,25,2}

SetVariable Smooth\_F,win=Main\_window, help={"The number of points in the smoothing window for Savitzky-Golay smoothing."}

endif

End

Function Additional\_Filters(ctrlName,checked) : CheckBoxControl

String ctrlName

Variable checked

SetDataFolder $"root:Quanta"

NVAR Smooth\_more=Smooth\_more

NVAR Smooth\_Derivative=Smooth\_Derivative

strswitch(ctrlName)

case "Scales\_Smooth\_Add":

If(checked==1)

Smooth\_more=1

else

Smooth\_more=0

endif

break

case "Scales\_Smooth\_Diff":

If(checked==1)

Smooth\_Derivative=1

else

Smooth\_Derivative=0

endif

break

endswitch

End

Function Gaussian\_Filter(Filtered\_Trace\_Name,Type)

String Filtered\_Trace\_Name

String Type

SetDataFolder $"root:Quanta"

Wave FFT\_wave=$Filtered\_Trace\_Name

Variable npnts= numpnts(FFT\_wave)

Variable Start\_X=pnt2x(FFT\_wave, 0)

Variable End\_X=pnt2x(FFT\_wave,npnts)

If(npnts>1e+6)

DoAlert 1, "During Fourier transforms of large traces (such as this one) the computer may freeze for a few seconds. As an alternative use Binomial smoothing.\rTo continue with Gaussian filter press 'Yes' or press 'No' to cancel."

If(V\_Flag!=1)

return 0

endif

endif

NVAR Smoothing\_Factor=Smoothing\_Factor

Variable Cutoff\_Amplitude=1/(sqrt(2)) // 50% power

Variable Gauss\_Width= Smoothing\_Factor/sqrt(-ln(Cutoff\_Amplitude))

Redimension/N=(npnts\*2) FFT\_wave // the wave has to have an EVEN number of rows

FFT FFT\_wave

WAVE/C FFT\_wave\_complex=FFT\_wave

If(cmpstr(Type,"Low")==0)

FFT\_wave\_complex\*=cmplx(exp(-(x^2/(Gauss\_Width^2))),0) // Low-pass Gaussian

else

FFT\_wave\_complex\*=cmplx(1-exp(-(x^2/(Gauss\_Width^2))),0) // High-pass Gaussian

endif

IFFT FFT\_wave

Redimension/N=(npnts) FFT\_wave

SetScale/I x Start\_X,End\_X,"s", FFT\_wave

End

Function Gaussian\_to\_Binomial\_Calc(Frequency\_Hz)

Variable Frequency\_Hz

SetDataFolder $"root:Quanta"

NVAR T\_Delta=T\_Delta

Variable Binomial\_coeff=0.02873\*(Frequency\_Hz \* T\_Delta/1000) ^(-2.0764)

Return max(round(Binomial\_coeff),1)

end

Function Binomial\_to\_Gaussian\_Calc(Binomial\_coeff)

Variable Binomial\_coeff

SetDataFolder $"root:Quanta"

NVAR T\_Delta=T\_Delta

Variable Frequency\_Hz=0.1809/(T\_Delta/1000)\*Binomial\_coeff ^(-0.4815)

Return trunc(Frequency\_Hz)

end

Function Filter\_Estimate(ctrlName) : ButtonControl

String ctrlName

SetDataFolder $"root:Quanta"

Wave Working\_trace\_copy=Working\_trace\_copy

Variable Point1,Point2,Freq

If(numpnts(Working\_trace\_copy)<=1)

print "No data loaded!"

abort

endif

Duplicate/O Working\_trace\_copy diff1\_nosmooth

Differentiate diff1\_nosmooth

WaveStats/Q diff1\_nosmooth

Variable dIMax=V\_max

FindLevel/Q /R=(V\_maxloc,0) diff1\_nosmooth,0

Point1=V\_LevelX

FindLevel/Q /R=(V\_maxloc,) diff1\_nosmooth,0

Point2=V\_LevelX

Variable Imax=Working\_trace\_copy(Point2)-Working\_trace\_copy(Point1)

Freq=dIMax/Imax/2

Duplicate/O Working\_trace\_copy diff1\_nosmooth

Variable Binomial\_coeff=Gaussian\_to\_Binomial\_Calc(Freq\*3)

Smooth Binomial\_coeff, diff1\_nosmooth

Differentiate diff1\_nosmooth

WaveStats/Q diff1\_nosmooth

dIMax=V\_max

FindLevel/Q /R=(V\_maxloc,0) diff1\_nosmooth,0

Point1=V\_LevelX

FindLevel/Q /R=(V\_maxloc,) diff1\_nosmooth,0

Point2=V\_LevelX

Freq=dIMax/Imax/2

MoveWindow/C 1,1,1,1

Print "\r"

String Formated\_String

sprintf Formated\_String, "%.0W1PHz", Freq

Print "Signal frequency is ~",Formated\_String

Print "Recommended filters:"

sprintf Formated\_String, "%.0W1PHz", 2\*Freq

Print "\tSignal smoothing: Binomial",Formated\_String

sprintf Formated\_String, "%.0W1PHz", Freq/2

Print "\tAdditional smoothing: Binomial",Formated\_String

sprintf Formated\_String, "%.0W1PHz", Freq

Print "\t1st derrivative smoothing: Binomial",Formated\_String

end

// Fall\_fit\_Extrap[i]=

// StringByKey("N", Fall\_fit\_Extrap[i]) Spike number in a series of separated spikes

// StringByKey("Total", Fall\_fit\_Extrap[i]) Total number of separated spikes

// StringByKey("Fit", Fall\_fit\_Extrap[i]) Function used to extrapolate the falling phase of the spike (Exp or Line)

// StringByKey("Tau", Fall\_fit\_Extrap[i]) Time constant of exponential decay (user-controlled). Equals -1 if linear fit is used.

// Also:

// ReplaceStringByKey("Fit", Fall\_fit\_Extrap[i], "Exp")

// HAPPY RECORDINGS!