Axel-hub

2.3.0.17

by Teodor Krastev for Imperial College London

1 Axel hub Introduction 1

1.0.1 Scan panel . . . 1
1.0.2 Axel-chart (top) . . . 2
1.0.3 Signal panel (middle) contains two charts: . . . 2
1.0.4 Fringes / Acceleration panel (bottom) . . . 2

2 Namespace Index 5

2.1 Packages . . . 5

3 Hierarchical Index 7

3.1 Class Hierarchy . . . 7

4 Class Index 9

4.1 Class List . . . 9

5 File Index 11

5.1 File List . . . 11

6 Namespace Documentation 13

6.1 Axel_hub Namespace Reference . . . 13
6.1.1 Function Documentation . . . 16
6.1.1.1 ActiveRemote() . . . 16
6.1.1.2 backMME() . . . 16
6.1.1.3 calcContrast() . . . 17
6.1.1.4 centreFringe() . . . 17
6.1.1.5 deconstructAccel() . . . 18
6.1.1.6 dispatcherTimer_Tick() . . . 19
6.1.1.7 fillReport() . . . 19
6.1.1.8 Flip() . . . 19
6.1.1.9 GetBufferSize() . . . 20
6.1.1.10 GetSamplingPeriod() . . . 20
6.1.1.11 Image_MouseDown() . . . 20
6.1.1.12 Init() . . . 20
6.1.1.13 InitOptions() . . . 21
6.1.1.14 LogEvent() . . . 21
6.1.1.15 LogHandler() . . . 21
6.1.1.16 nextShot() . . . 22
6.1.1.17 OnActiveComm() . . . 22
6.1.1.18 OnAsyncSend() . . . 22
6.1.1.19 OnJumboRepeat() . . . 23
6.1.1.20 OnRealSampling() . . . 23
6.1.1.21 OnReceive() . . . 23
6.1.1.22 OpenConfigFile() . . . 24
6.1.1.23 PID() . . . 24

6.1.1.24 RemoteEvent() . . . 24
6.1.1.25 RemoteHandler() . . . 24
6.1.1.26 RemoteModeEvent() . . . 25
6.1.1.27 Reset() . . . 25
6.1.1.28 SaveConfigFile() . . . 25
6.1.1.29 scanClass() . . . 25
6.1.1.30 SendJson() . . . 25
6.1.1.31 SetActivity() . . . 26
6.1.1.32 SetFringeParams() . . . 26
6.1.1.33 SetSamplingRate() . . . 26
6.1.1.34 StartDelegate() . . . 27
6.1.1.35 StartEvent() . . . 27
6.1.1.36 StartHandler() . . . 27
6.1.1.37 Status() . . . 27
6.1.1.38 strobesUC() . . . 27
6.1.1.39 UpdateModes() . . . 28
6.1.1.40 UserControl_Loaded() . . . 28
6.1.1.41 zeroFringe() . . . 28
6.1.2 Variable Documentation . . . 28
6.1.2.1 _jumboButton . . . 28
6.1.2.2 _PID_Enabled . . . 29
6.1.2.3 _Running . . . 29
6.1.2.4 ArrangedPartner . . . 29
6.1.2.5 configFile . . . 29
6.1.2.6 currentTime . . . 29
6.1.2.7 Down . . . 29
6.1.2.8 dStack . . . 30
6.1.2.9 dTimer . . . 30
6.1.2.10 genOptions . . . 30
6.1.2.11 grpMME . . . 30
6.1.2.12 iStack . . . 30
6.1.2.13 jumboButton . . . 30
6.1.2.14 lastContrast . . . 31
6.1.2.15 lastMMEin . . . 31
6.1.2.16 lastMMEout . . . 31
6.1.2.17 logger . . . 31
6.1.2.18 Low . . . 31
6.1.2.19 OnActiveRemote . . . 31
6.1.2.20 OnLog . . . 32
6.1.2.21 OnRemote . . . 32
6.1.2.22 OnRemoteMode . . . 32
6.1.2.23 OnStart . . . 32

6.1.2.24 PID_Enabled . . . 32
6.1.2.25 realSampling . . . 33
6.1.2.26 remote . . . 33
6.1.2.27 remoteMode . . . 33
6.1.2.28 runI . . . 33
6.1.2.29 Running . . . 33
6.1.2.30 scanModes . . . 34
6.1.2.31 set . . . 34
6.1.2.32 Titles . . . 34
6.1.2.33 totalTime . . . 34
6.1.2.34 Up . . . 34
6.2 Axel_hub::Properties Namespace Reference . . . 34
6.3 OptionsNS Namespace Reference . . . 35
6.3.1 Enumeration Type Documentation . . . 35
6.3.1.1 RemoteMode . . . 35
6.4 XamlGeneratedNamespace Namespace Reference . . . 35

7 Class Documentation **37**

7.1 Axel_hub.accelCalibr Struct Reference . . . 37
7.1.1 Detailed Description . . . 37
7.1.2 Member Function Documentation . . . 37
7.1.2.1 accel() . . . 37
7.1.3 Member Data Documentation . . . 38
7.1.3.1 cK0 . . . 38
7.1.3.2 cK1 . . . 38
7.1.3.3 model . . . 38
7.1.3.4 pK0 . . . 38
7.1.3.5 pK1 . . . 39
7.1.3.6 rAccel . . . 39
7.1.3.7 rTemper . . . 39
7.1.3.8 SN . . . 39
7.2 Axel_hub::App Class Reference . . . 39
7.2.1 Detailed Description . . . 40
7.3 Axel_hub.AxelAxesClass Class Reference . . . 40
7.3.1 Detailed Description . . . 41
7.3.2 Constructor & Destructor Documentation . . . 41
7.3.2.1 AxelAxesClass() . . . 41
7.3.3 Member Function Documentation . . . 42
7.3.3.1 AddAxis() . . . 42
7.3.3.2 byName() . . . 42
7.3.3.3 Clear() . . . 43
7.3.3.4 Closing() . . . 43

7.3.3.5 DoAcquire() . . . 43
7.3.3.6 DoAcquireTemperature() . . . 44
7.3.3.7 DoJumboScan() . . . 44
7.3.3.8 DoRemote() . . . 44
7.3.3.9 jumboRepeat() . . . 45
7.3.3.10 LogEvent() . . . 45
7.3.3.11 LogHandler() . . . 45
7.3.3.12 prfIdx() . . . 45
7.3.3.13 SaveDefaultModes() . . . 46
7.3.3.14 set2startADC24() . . . 46
7.3.3.15 SetChartStrobes() . . . 46
7.3.3.16 startADC() . . . 47
7.3.3.17 UpdateFromOptions() . . . 47
7.3.4 Member Data Documentation . . . 47
7.3.4.1 axelMems . . . 47
7.3.5 Property Documentation . . . 47
7.3.5.1 memsRunning . . . 48
7.3.5.2 rCount . . . 48
7.3.6 Event Documentation . . . 48
7.3.6.1 OnLog . . . 48
7.4 Axel_hub::AxelAxisClass Class Reference . . . 48
7.4.1 Detailed Description . . . 49
7.5 Axel_hub::AxelChart Class Reference . . . 49
7.5.1 Detailed Description . . . 49
7.6 Axel_hub::AxelChartClass Class Reference . . . 49
7.6.1 Detailed Description . . . 50
7.7 Axel_hub.AxelMems Class Reference . . . 50
7.7.1 Detailed Description . . . 51
7.7.2 Member Enumeration Documentation . . . 51
7.7.2.1 TimingModes . . . 51
7.7.3 Constructor & Destructor Documentation . . . 52
7.7.3.1 AxelMems() . . . 52
7.7.4 Member Function Documentation . . . 52
7.7.4.1 AcquireEvent() . . . 52
7.7.4.2 AcquireHandler() . . . 52
7.7.4.3 configureVITask() . . . 53
7.7.4.4 isDevicePlugged() . . . 53
7.7.4.5 readBurst() . . . 53
7.7.4.6 RealConvRate() . . . 53
7.7.4.7 RealSamplingEvent() . . . 54
7.7.4.8 RealSamplingHandler() . . . 54
7.7.4.9 Reset() . . . 54

7.7.4.10 SetStopwatch() . . . 54
7.7.4.11 StartAcquisition() . . . 54
7.7.4.12 StartStopwatch() . . . 55
7.7.4.13 StopAcquisition() . . . 55
7.7.4.14 TimeElapsed() . . . 55
7.7.5 Member Data Documentation . . . 55
7.7.5.1 AdjustTimelineToStopwatch . . . 55
7.7.5.2 FixConvRate . . . 56
7.7.5.3 hw . . . 56
7.7.5.4 memsX . . . 56
7.7.5.5 rawData . . . 56
7.7.5.6 Timeout . . . 56
7.7.5.7 TimingMode . . . 56
7.7.6 Property Documentation . . . 57
7.7.6.1 activeChannel . . . 57
7.7.6.2 nSamples . . . 57
7.7.6.3 running . . . 57
7.7.6.4 sampleRate . . . 57
7.7.7 Event Documentation . . . 57
7.7.7.1 OnAcquire . . . 57
7.7.7.2 OnRealSampling . . . 58
7.8 Axel_hub.AxelMemsTemperature Class Reference . . . 58
7.8.1 Detailed Description . . . 58
7.8.2 Constructor & Destructor Documentation . . . 58
7.8.2.1 AxelMemsTemperature() . . . 58
7.8.3 Member Function Documentation . . . 58
7.8.3.1 TakeTheTemperature() . . . 59
7.8.4 Member Data Documentation . . . 59
7.8.4.1 hw . . . 59
7.9 Axel_hub.DataStack Class Reference . . . 59
7.9.1 Detailed Description . . . 61
7.9.2 Constructor & Destructor Documentation . . . 61
7.9.2.1 DataStack() . . . 61
7.9.3 Member Function Documentation . . . 62
7.9.3.1 Add() . . . 62
7.9.3.2 AddPoint() . . . 62
7.9.3.3 AddRange() . . . 63
7.9.3.4 Clear() . . . 63
7.9.3.5 Clone() . . . 63
7.9.3.6 Compress() . . . 64
7.9.3.7 CopyEach() . . . 64
7.9.3.8 ExportToArray() . . . 64

7.9.3.9 fillSamples() . . . 65
7.9.3.10 Fit2Limit() . . . 65
7.9.3.11 ImportFromArray() . . . 65
7.9.3.12 importFromArrays() . . . 66
7.9.3.13 indexByX() . . . 66
7.9.3.14 OpenPair() . . . 66
7.9.3.15 pointSDev() . . . 67
7.9.3.16 pointXs() . . . 67
7.9.3.17 pointYs() . . . 68
7.9.3.18 Portion() . . . 68
7.9.3.19 RefreshEvent() . . . 68
7.9.3.20 RefreshHandler() . . . 68
7.9.3.21 Rescale() . . . 69
7.9.3.22 SavePair() . . . 70
7.9.3.23 statsByIdx() . . . 70
7.9.3.24 statsByTime() . . . 71
7.9.3.25 TimePortion() . . . 71
7.9.4 Member Data Documentation . . . 72
7.9.4.1 generalIdx . . . 72
7.9.4.2 logger . . . 72
7.9.4.3 maxDepth . . . 72
7.9.4.4 RefFileStats . . . 72
7.9.4.5 stopWatch . . . 72
7.9.4.6 visualCountLimit . . . 73
7.9.5 Property Documentation . . . 73
7.9.5.1 Depth . . . 73
7.9.5.2 First . . . 73
7.9.5.3 Last . . . 73
7.9.5.4 lastError . . . 73
7.9.5.5 prefix . . . 74
7.9.5.6 rem . . . 74
7.9.5.7 Running . . . 74
7.9.5.8 StackMode . . . 74
7.9.5.9 TimeSeriesMode . . . 74
7.9.6 Event Documentation . . . 74
7.9.6.1 OnRefresh . . . 75
7.10 Axel_hub::FringeParams Struct Reference . . . 75
7.10.1 Detailed Description . . . 75
7.10.2 Member Data Documentation . . . 75
7.10.2.1 offset . . . 75
7.10.2.2 period . . . 76
7.10.2.3 phase . . . 76

7.11 OptionsNS.GeneralOptions Class Reference . . . 76
7.11.1 Detailed Description . . . 77
7.11.2 Member Enumeration Documentation . . . 77
7.11.2.1 SaveModes . . . 77
7.11.3 Member Function Documentation . . . 77
7.11.3.1 Save() . . . 78
7.11.4 Member Data Documentation . . . 78
7.11.4.1 saveModes . . . 78
7.11.5 Property Documentation . . . 78
7.11.5.1 AxesChannels . . . 78
7.11.5.2 followPID . . . 78
7.11.5.3 intN2 . . . 78
7.11.5.4 JumboRepeat . . . 79
7.11.5.5 JumboScan . . . 79
7.11.5.6 LogFilePrec . . . 79
7.11.5.7 Mems2SignDelay . . . 79
7.11.5.8 Mems2SignLen . . . 79
7.11.5.9 MemsHw . . . 79
7.11.5.10 MemsInJumbo . . . 80
7.11.5.11 RawSignalAvg . . . 80
7.11.5.12 SaveFilePrec . . . 80
7.11.5.13 saveVisuals . . . 80
7.11.5.14 ShowMemsIfRunning . . . 80
7.11.5.15 SignalCursorPrec . . . 80
7.11.5.16 SignalTablePrec . . . 81
7.11.5.17 TemperatureCompensation . . . 81
7.11.5.18 TemperatureEnabled . . . 81
7.11.5.19 TemperatureHw . . . 81
7.11.5.20 TrendSignalLen . . . 81
7.12 XamlGeneratedNamespace::GeneratedInternalTypeHelper Class Reference . . . 82
7.12.1 Detailed Description . . . 82
7.13 Axel_hub::JoinOptimClass Class Reference . . . 82
7.13.1 Detailed Description . . . 82
7.14 Axel_hub::MainWindow Class Reference . . . 83
7.14.1 Detailed Description . . . 83
7.15 OptionsNS.Modes Class Reference . . . 83
7.15.1 Detailed Description . . . 84
7.15.2 Member Function Documentation . . . 84
7.15.2.1 Save() . . . 85
7.15.3 Property Documentation . . . 85
7.15.3.1 AutoScaleBottom . . . 85
7.15.3.2 AutoScaleMiddle . . . 85

7.15.3.3 Background . . . 85
7.15.3.4 ChartUpdate . . . 85
7.15.3.5 DarkCurrent . . . 85
7.15.3.6 DoubleStrobe . . . 86
7.15.3.7 JoinLog . . . 86
7.15.3.8 JumboBy . . . 86
7.15.3.9 JumboCycles . . . 86
7.15.3.10 JumboFrom . . . 86
7.15.3.11 JumboTo . . . 86
7.15.3.12 Kcoeff . . . 87
7.15.3.13 kD . . . 87
7.15.3.14 kI . . . 87
7.15.3.15 kP . . . 87
7.15.3.16 MemsEnabled . . . 87
7.15.3.17 MiddleFrame . . . 87
7.15.3.18 N1 . . . 88
7.15.3.19 N2 . . . 88
7.15.3.20 Ntot . . . 88
7.15.3.21 offset . . . 88
7.15.3.22 phi0 . . . 88
7.15.3.23 PID_Enabled . . . 88
7.15.3.24 PowerCoeff . . . 89
7.15.3.25 RN1 . . . 89
7.15.3.26 RN2 . . . 89
7.15.3.27 RollMean . . . 89
7.15.3.28 RsltChrtUpdate . . . 89
7.15.3.29 RsltTblUpdate . . . 89
7.15.3.30 scale . . . 90
7.15.3.31 ShowFreq . . . 90
7.15.3.32 SignalLog . . . 90
7.15.3.33 StackDepth . . . 90
7.15.3.34 StdDev . . . 90
7.15.3.35 TblUpdate . . . 90
7.15.3.36 TopFrame . . . 91
7.15.3.37 TopOfTopFrame . . . 91
7.16 OptionsNS::OptionsWindow Class Reference . . . 91
7.16.1 Detailed Description . . . 91
7.17 Axel_hub::Properties::Resources Class Reference . . . 91
7.17.1 Detailed Description . . . 92
7.18 Axel_hub::scanClass Class Reference . . . 92
7.18.1 Detailed Description . . . 92
7.19 OptionsNS.ScanModes Class Reference . . . 92

7.19.1 Detailed Description . . . 93
7.19.2 Member Function Documentation . . . 93
7.19.2.1 Save() . . . 93
7.19.3 Member Data Documentation . . . 93
7.19.3.1 remoteMode . . . 93
7.19.4 Property Documentation . . . 93
7.19.4.1 Height . . . 94
7.19.4.2 Left . . . 94
7.19.4.3 SamplingFreq . . . 94
7.19.4.4 SizeLimit . . . 94
7.19.4.5 SizeLimitMode . . . 94
7.19.4.6 TimeLimit . . . 94
7.19.4.7 TimeLimitMode . . . 95
7.19.4.8 Top . . . 95
7.19.4.9 Width . . . 95
7.20 Axel_hub::Properties::Settings Class Reference . . . 95
7.20.1 Detailed Description . . . 95
7.21 Axel_hub.ShotList Class Reference . . . 96
7.21.1 Detailed Description . . . 96
7.21.2 Constructor & Destructor Documentation . . . 97
7.21.2.1 ShotList() . . . 97
7.21.3 Member Function Documentation . . . 97
7.21.3.1 Add() . . . 97
7.21.3.2 archiScan() . . . 97
7.21.3.3 resetScan() . . . 98
7.21.3.4 Save() . . . 98
7.21.4 Member Data Documentation . . . 98
7.21.4.1 conditions . . . 98
7.21.4.2 depth . . . 99
7.21.5 Property Documentation . . . 99
7.21.5.1 archiveMode . . . 99
7.21.5.2 enabled . . . 99
7.21.5.3 FileCount . . . 99
7.21.5.4 filename . . . 99
7.21.5.5 lastIdx . . . 100
7.21.5.6 savingMode . . . 100
7.22 Axel_hub::signalClass Class Reference . . . 100
7.22.1 Detailed Description . . . 100
7.23 Axel_hub.SingleShot Class Reference . . . 100
7.23.1 Detailed Description . . . 101
7.23.2 Constructor & Destructor Documentation . . . 101
7.23.2.1 SingleShot() [1/4] . . . 101

7.23.2.2 SingleShot() [2/4] . . . 102
7.23.2.3 SingleShot() [3/4] . . . 102
7.23.2.4 SingleShot() [4/4] . . . 102
7.23.3 Member Function Documentation . . . 102
7.23.3.1 deconstructAccel() . . . 102
7.23.3.2 idxByTime() . . . 103
7.23.3.3 IsEmpty() . . . 103
7.23.3.4 memsPortion() . . . 103
7.23.3.5 memsWeightAccel() . . . 104
7.23.4 Member Data Documentation . . . 104
7.23.4.1 precision . . . 104
7.23.4.2 quant . . . 104
7.23.5 Property Documentation . . . 104
7.23.5.1 AsString . . . 105
7.23.5.2 mems . . . 105
7.24 Axel_hub::strobesUC Class Reference . . . 105
7.24.1 Detailed Description . . . 105

8 File Documentation **107**

8.1 App.g.cs File Reference . . . 107
8.2 App.g.i.cs File Reference . . . 107
8.3 App.xaml.cs File Reference . . . 107
8.4 AssemblyInfo.cs File Reference . . . 108
8.5 Axel-hub_Content.g.i.cs File Reference . . . 108
8.6 AxelAxes.cs File Reference . . . 108
8.7 AxelAxisUC.g.cs File Reference . . . 108
8.8 AxelAxisUC.g.i.cs File Reference . . . 108
8.9 AxelAxisUC.xaml.cs File Reference . . . 109
8.10 AxelChart.g.i.cs File Reference . . . 109
8.11 AxelChartUC.g.cs File Reference . . . 109
8.12 AxelChartUC.g.i.cs File Reference . . . 109
8.13 AxelChartUC.xaml.cs File Reference . . . 110
8.14 AxelHMems.cs File Reference . . . 110
8.15 DataPrimitives.cs File Reference . . . 110
8.16 DataStackLib.cs File Reference . . . 111
8.17 GeneratedInternalTypeHelper.g.cs File Reference . . . 111
8.18 GeneratedInternalTypeHelper.g.i.cs File Reference . . . 111
8.19 JoinOptimUC.g.i.cs File Reference . . . 111
8.20 MainWindow.g.cs File Reference . . . 111
8.21 MainWindow.g.i.cs File Reference . . . 112
8.22 MainWindow.xaml.cs File Reference . . . 112
8.23 Options.g.cs File Reference . . . 112

8.24 Options.g.i.cs File Reference . . . 113
8.25 Options.g.i.cs File Reference . . . 113
8.26 Options.xaml.cs File Reference . . . 113
8.27 OptionsType.cs File Reference . . . 113
8.28 README.md File Reference . . . 114
8.29 Resources.Designer.cs File Reference . . . 114
8.30 scanUC.g.cs File Reference . . . 114
8.31 scanUC.g.i.cs File Reference . . . 114
8.32 scanUC.xaml.cs File Reference . . . 115
8.33 Settings.Designer.cs File Reference . . . 116
8.34 signalUC.g.cs File Reference . . . 117
8.35 signalUC.g.i.cs File Reference . . . 117
8.36 signalUC.xaml.cs File Reference . . . 117
8.37 strobeControlUC.g.i.cs File Reference . . . 117
8.38 strobesUC.g.cs File Reference . . . 118
8.39 strobesUC.g.i.cs File Reference . . . 118
8.40 strobesUC.xaml.cs File Reference . . . 118

Index **121**

Chapter 1

Axel hub Introduction

One way to make a software system more reliable is to distribute resources thru so called (loose coupling), so the central piece of software (MOTmaster) would not be affected by eventual bottleneck in some of visualization or data processing parts (Axel-hub) of the system. Axel hub is the principal data visualization, logging and processing hub for the quantum accelerometer. It receives raw data from MotMaster2 thru specially designed fast communication channel, visualize the signal (raw data), chart and log its trend. Another major part of Axel Hub is the control of MEMS measurements via 24 bit ADC (NI 9251). Axel hub application can run independently from MOTmaster, but its main purpose by design is to work in tandem with MOTmaster providing visualization and some data processing features for the Navigator experiment. For the communication between these two a special communication channel has been written with speed in mind. The communication channel uses customizable part Windows messaging system and combines with JSON type of communication protocol described (as known internally) in (The Book of JaSON). The average transmitting time for a message (command) including interpretation is under 1 ms. For the aimed navigator cycle period of 100 ms, it is a less than 1 percent. Visually Axel-hub has 5 panels resizable by splitters. The panels represent different functionalities in groups.

1.0.1 Scan panel

MEMS measurements are controllable by the top left panel. The actual control is on the ADC24 which 24 bit analogue-to-digital converter (National Instruments NI 9251). The user can set continuous or finite measurement with desired sampling frequency. The actual sampling frequency would be one from a list with pre-set frequencies. A buffer size can be set in seconds (Time Limit tab) or in number of points (Size limit tab). The buffer size will be the number of point measured in finite mode or the number of point taken at one shot in continuous mode. In later mode, there are no gaps in time between shots and the visualization is updated with every shot. Once the conditions for the measurement are set the user can start measuring by pressing the Start button. When Remote tab is active Axel hub acts as visualization/log hub for MOTmaster. In this case when the measurement is initiated by MOTmaster Axel-hub passively will show the data on corresponding chart. In another way to proceed is to execute a procedure (called Jumbo) from Axel hub. The first phase is Jumbo scan where Axel hub requires MOTmaster to make a scan of the interferometric fringes and shows the fringes in the lower panel on the right. Once the scan is finished Axel hub will ask the user to place one or two cursors on the side(s) of best fringe. After that the second phase (Jumbo repeat) will follow the movement of the fringe by the intensity of the signal at side(s) of the fringe. The algorithm of following is proportional:integral:derivative (PID) based. The log panel (left bottom) provides flexible ways to show communication command/data flow. In default mode, the log window will list only small (most informative) portion of each command/shot. In verbatim mode the complete data flow will be visible, but this could slow down the Axel-hub performance.

1.0.2 Axel-chart (top)

Axel-chart is a (user control) in terms of Visual Studio and it is a visual panel separated in two (the top two charts in the image). During a measurement, the top part shows incoming data, usually a relatively small porting of it. The idea is to be quick and check only for obvious inconsistencies of incoming data flow. When it is used offline, the panel allows the user to browse large data in convenient manner (piece by piece). The lower part consists of three charts and panel with tools for file and statistics. The first chart is Overview and it shows the whole spectrum so it could be sluggish if the size is bigger than 10k. Some tools for visual manipulation (and copy the pic) are available on this tab (as on others too). The second tab is Histogram: it calculates a histogram of the spectrum and optionally the user can fit a Gaussian curve over the data. A useful feature here is Window mode (shown on Axel-hub image) which allow the user to select a portion of the histogram (when histogram is multi-mode) and fit the curve over only the selected part.

The last tab provides number of features:

- file operations (Open and Save) including a remark for particular measurement description.
- Chart options as Xaxis units, and some others, including the maximum number of points kept (depth)
- Calculating and displaying the current value and dispersion of MEMS measurement as taken for the last (Time slice)
- Split data is designed to be used two level signal (as from an optical chopper) and split the spectrum in two: upper part and lower part.
- Extract part, it extracts the visual part of the top chart and creates a new spectrum from it

1.0.3 Signal panel (middle) contains two charts:

- The right one shows the optical signal as it comes from photodiode detector of the experiment
- The left chart provides the trends of N1, N2, N.total, N1.relative, and N2.relative; individually switchable.
- The user can optionally correct for dark current or background as well as show/hide the standard deviations of the measurement The last tab (Opt/Stat) provides file operations (Open and Save) for the trends chart series.

1.0.4 Fringes / Acceleration panel (bottom)

This panel has two tabs with two charts (one each). The Fringes tab/chart provides visualization in case of MO↩ Tmaster scan, initiated either by MOTmaster (simple scan) or by Axel-hub (jumbo-scan). Accel.trend serves similar to Fringes purpose except it is for repeat, respectively repeat initiated by MOTmaster is called (simple repeat) and by Axel-hub: jumbo-repeat. On the right are proportional.integral.derivative (PID) controller parameters controlling the phase correction extracted from intensity of the signal on the side(s) of the chosen fringe by PID. On the last tab (Opt/Stats) the features are:

- Jumbo procedure setting, as range and step for fringe scan and number of repeat cycles (negative value for continue)
- file operations (Open and Save) for Fringes data
- vibrations analysis, use the Navigator system to detect vibrations from the environment, mostly for testing the condition of the experiment

In conclusion, the software provides:

- The MEMS (classical) acceleration measurements could be done independently (stay alone) or in synchronization with MotMaster2 data flow. In any case the data can be chart (short term and long term) and histogram or FFT charts could be calculated and drawn in real time. Some tools for off line data processing are available too.

- The middle panel provides visualization of raw MotMaster2 data (the signal) and some signal trends (N1, N2, Ntot, rel.N1 and rel.N2).
- There are two major operating modes: Simple (Axel Hub is in slave position) and Jumbo (MotMaster2 is in slave position).
- In Simple operation mode the bottom panel is used for charting results: in scan mode: Fringes tab or repeat mode: Accel.Trend tab.
- Another major feature is so called Jumbo mode, in this mode MotMaster2 is under Axel hub control providing first fringes pattern (scan of Raman phase) and then using PID algorithm following a chosen fringe position (pi flip procedure) in order to calculated the quantum acceleration.
- Finally, the quantum and the classical acceleration measurements are combined in one result acceleration value. All the accelerations are presented in a chart (Acce.Trend) and in a table.
- Optionally some of the intermediate results can be logged for later adjusting the processing parameters.

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

Axel_hub 13
Axel_hub::Properties 34
OptionsNS 35
XamlGeneratedNamespace 35

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Axel_hub.accelCalibr . . . 37
Application
  Axel_hub::App . . . 39
  Axel_hub::App . . . 39
  Axel_hub::App . . . 39
ApplicationSettingsBase
  Axel_hub::Properties::Settings . . . 95
Axel_hub.AxelMems . . . 50
Axel_hub.AxelMemsTemperature . . . 58
Axel_hub::FringeParams . . . 75
OptionsNS.GeneralOptions . . . 76
IComponentConnector
  Axel_hub::AxelAxisClass . . . 48
  Axel_hub::AxelAxisClass . . . 48
  Axel_hub::AxelChart . . . 49
  Axel_hub::AxelChartClass . . . 49
  Axel_hub::AxelChartClass . . . 49
  Axel_hub::JoinOptimClass . . . 82
  Axel_hub::MainWindow . . . 83
  Axel_hub::MainWindow . . . 83
  Axel_hub::scanClass . . . 92
  Axel_hub::scanClass . . . 92
  Axel_hub::signalClass . . . 100
  Axel_hub::signalClass . . . 100
  Axel_hub::strobesUC . . . 105
  Axel_hub::strobesUC . . . 105
  Axel_hub::strobesUC . . . 105
  OptionsNS::OptionsWindow . . . 91
  OptionsNS::OptionsWindow . . . 91
  OptionsNS::OptionsWindow . . . 91
InternalTypeHelper
  XamlGeneratedNamespace::GeneratedInternalTypeHelper . . . 82
List
  Axel_hub.AxelAxesClass . . . 40
  Axel_hub.DataStack . . . 59

Axel_hub.ShotList 96
OptionsNS.Modes 83
Axel_hub::Properties::Resources 91
OptionsNS.ScanModes 92
Axel_hub.SingleShot 100
UserControl
Axel_hub::AxelAxisClass 48
Axel_hub::AxelAxisClass 48
Axel_hub::AxelChart 49
Axel_hub::AxelChartClass 49
Axel_hub::AxelChartClass 49
Axel_hub::JoinOptimClass 82
Axel_hub::scanClass 92
Axel_hub::scanClass 92
Axel_hub::signalClass 100
Axel_hub::signalClass 100
Axel_hub::strobesUC 105
Axel_hub::strobesUC 105
Axel_hub::strobesUC 105
UserControl
Axel_hub::AxelAxisClass 48
Axel_hub::AxelChartClass 49
Axel_hub::signalClass 100
Window
Axel_hub::MainWindow 83
Axel_hub::MainWindow 83
Axel_hub::MainWindow 83
OptionsNS::OptionsWindow 91
OptionsNS::OptionsWindow 91
OptionsNS::OptionsWindow 91
OptionsNS::OptionsWindow 91

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Axel_hub.accelCalibr
: Acceleration calibration with optional temperature compensation particular to each MEMS device 37

Axel_hub::App
: Interaction logic for App.xaml . . . 39

Axel_hub.AxelAxesClass
: Intermediator between incomming data flow from ucScan user component and AxelAxis user components . . . 40

Axel_hub::AxelAxisClass
: Interaction logic for AxelAxisUC.xaml AxelAxisClass repressents a single axis of acceleration encapsulated and accesable in AxelAxes list of AxelAxisClass Future intermediator abstract movement (linear or rotation) component will be implemented . . . 48

Axel_hub::AxelChart
: AxelChart . . . 49

Axel_hub::AxelChartClass
: Interaction logic for AxelChart.xaml . . . 49

Axel_hub.AxelMems
: The hardware abstraction for MEMS with ADC24 (NI9251) device . . . 50

Axel_hub.AxelMemsTemperature
: The temperature in a class abstraction . . . 58

Axel_hub.DataStack
: You (developer) need to set TimeMode and one of SizeLimit or TimeLimit TimeMode is about the way DataStack limits its size The output is from standart List method ToArray in order to set DataSource of Graph . . . 59

Axel_hub::FringeParams
: fringes(phi) = cos(period ∗ phi + phase) + offset . . . 75

OptionsNS.GeneralOptions
: general options from Options dialog window accesable everywhere . . . 76

XamlGeneratedNamespace::GeneratedInternalTypeHelper
: GeneratedInternalTypeHelper . . . 82

Axel_hub::JoinOptimClass
: JoinOptimClass . . . 82

Axel_hub::MainWindow
: Interaction logic for MainWindow.xaml command line arguments (space separated): -remote:`partner` -hw:`config.file` where `partner` is remote partner name `title`; hw<c>hardware, `config.file.hw` is in Config folder . . . 83

OptionsNS.Modes
Visuals and prameters for Top: Axel-chart Middle: Signal panel charts Bottom: Scan and Accel trend tabs/charts . . . 83
OptionsNS::OptionsWindow
OptionsWindow . . . 91
Axel_hub::Properties::Resources
A strongly-typed resource class, for looking up localized strings, etc . . . 91
Axel_hub::scanClass
scanClass . . . 92
OptionsNS.ScanModes
visuals for the app, MEMS aqcuisition params and scan modes . . . 92
Axel_hub::Properties::Settings . . . 95
Axel_hub.ShotList
List / series of single shots . . . 96
Axel_hub::signalClass
signalClass . . . 100
Axel_hub.SingleShot
Class representing single shot with both components quant (MOT) and MEMS (ADC24) . . . 100
Axel_hub::strobesUC
strobesUC . . . 105

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

App.g.cs . . . 107
App.g.i.cs . . . 107
App.xaml.cs . . . 107
AssemblyInfo.cs . . . 108
Axel-hub_Content.g.i.cs . . . 108
AxelAxes.cs . . . 108
AxelAxisUC.g.cs . . . 108
AxelAxisUC.g.i.cs . . . 108
AxelAxisUC.xaml.cs . . . 109
AxelChart.g.i.cs . . . 109
AxelChartUC.g.cs . . . 109
AxelChartUC.g.i.cs . . . 109
AxelChartUC.xaml.cs . . . 110
AxelHMems.cs . . . 110
DataPrimitives.cs . . . 110
DataStackLib.cs . . . 111
GeneratedInternalTypeHelper.g.cs . . . 111
GeneratedInternalTypeHelper.g.i.cs . . . 111
JoinOptimUC.g.i.cs . . . 111
MainWindow.g.cs . . . 111
MainWindow.g.i.cs . . . 112
MainWindow.xaml.cs . . . 112
Options.g.cs . . . 112
Options/Options.g.i.cs . . . 113
Options.g.i.cs . . . 113
Options.xaml.cs . . . 113
OptionsType.cs . . . 113
Resources.Designer.cs . . . 114
scanUC.g.cs . . . 114
scanUC.g.i.cs . . . 114
scanUC.xaml.cs . . . 115
Settings.Designer.cs . . . 116
signalUC.g.cs . . . 117
signalUC.g.i.cs . . . 117
signalUC.xaml.cs . . . 117

strobeControlUC.g.i.cs 117
strobesUC.g.cs 118
strobesUC.g.i.cs 118
strobesUC.xaml.cs 118

Chapter 6

Namespace Documentation

6.1 Axel_hub Namespace Reference

Namespaces

- Properties

Classes

- struct accelCalibr

  Acceleration calibration with optional temperature compensation particular to each MEMS device
- class App

  Interaction logic for App.xaml
- class AxelAxesClass

  Intermediator between incomming data flow from ucScan user component and AxelAxis user components
- class AxelAxisClass

  Interaction logic for AxelAxisUC.xaml AxelAxisClass represents a single axis of acceleration encapsulated and accesable in AxelAxes list of AxelAxisClass Future intermediator abstract movement (linear or rotation) component will be implemented.
- class AxelChart

  AxelChart
- class AxelChartClass

  Interaction logic for AxelChart.xaml
- class AxelMems

  The hardware abstraction for MEMS with ADC24 (NI9251) device
- class AxelMemsTemperature

  The temperature in a class abstraction
- class **calcAccel**

  Library for calculating acceleration from fringes, phase, etc
- class DataStack

  You (developer) need to set TimeMode and one of SizeLimit or TimeLimit TimeMode is about the way DataStack limits its size The output is from standart List method ToArray in order to set DataSource of Graph
- struct FringeParams

  fringes(phi) = cos(period ∗ phi + phase) + offset
- class JoinOptimClass

JoinOptimClass

- class MainWindow

  Interaction logic for MainWindow.xaml command line arguments (space separated): -remote:`partner` -hw:`config.file` where `partner` is remote partner name `title`; hw<c>hardware, `config.file.hw` is in Config folder

- class **MMDataConverter**

  Averaging the photo diode signals {"N2", "NTot", "B2", "BTot", "Bg"}

- class scanClass

  scanClass

- class ShotList

  List / series of single shots

- class signalClass

  signalClass

- class SingleShot

  Class representing single shot with both components quant (MOT) and MEMS (ADC24)

- class strobesUC

  strobesUC

Functions

- public delegate void StartDelegate ()
- public scanClass ()

  Class constructor - set defaults

- public void InitOptions (ref GeneralOptions _genOptions, ref ScanModes _scanModes)

  Initialize - set genOptions

- public void UpdateModes ()

  Set internal from visual modes

- public bool SendJson (string json, bool async=false)

  Wrapper of remote.sendCommand

- public void SetActivity (string act)
- public void SetSamplingRate (int rate)
- public void SetFringeParams (FringeParams fp)

  Show fringes params

- public void OnRealSampling (double _realSampling)
- private void dispatcherTimer_Tick (object sender, EventArgs e)

  Shows visual progress of ADC24 acquisition

- private void Status (string sts)
- protected void RemoteModeEvent (RemoteMode oldMode, RemoteMode newMode)
- private bool OnReceive (string message)

  Incomming from MM2/Axel-probe message

- public delegate void StartHandler (bool jumbo, bool down, double period, int sizeLimit)

  Start/Stop group operation wity ADC24 params

- protected void StartEvent (bool jumbo, bool down, double period, int sizeLimit)
- public delegate void RemoteHandler (string msg)

  Incomming message event thingy

- protected void RemoteEvent (string msg)
- public delegate void LogHandler (string txt, Color? clr=null)

  Log into left text box

- protected void LogEvent (string txt, Color? clr=null)
- protected void OnAsyncSend (bool OK, string json2send)

  Report sent message in log

- private void Image_MouseDown (object sender, MouseButtonEventArgs e)
- public double GetSamplingPeriod ()

  Get the sampling period regardless the units
- public int GetBufferSize ()

  Get the buffer size depending of settings
- protected void ActiveRemote (bool active)
- private void OnActiveComm (bool active, bool forced)

  Event when the connection goes ON/OFF
- private void UserControl_Loaded (object sender, RoutedEventArgs e)

  Some secondary to contructor initialilzations
- public strobesUC ()

  Class constructor
- public void Reset ()

  Initaile strobe for axel-probe simulated fringes
- public void Flip ()

  Exchange UP/DOWN strobe positions
- public void Init (string _prefix)

  Initiate strobe from file settings
- public void OnJumboRepeat (double _fringeScale, double _fringeShift, MMexec _grpMME, double contrastV)

  Call this before each Jumbo Repeat for group MMexec and modes synchronization
- public double centreFringe ()

  Calculating fringe centre
- public double calcContrast (double A)

  Calculating contrast
- public double zeroFringe ()

  Calculating zeroFringe - similar to centreFring but woth phase shift compensation
- public Dictionary< string, double > deconstructAccel (double accel, double mems)

  Deconstructing accaleration to acceleration components - see dictionary keys
- public double nextShot (int runID, double asymmetry, out double correction)

  Calculated phaseCorr - corrected Raman phase (0 if not PID)
- public MMexec backMME (int runID, double asymmetry, MMexec mme=null)

  Prepare back message with new Raman phase value
- private void fillReport (Dictionary< string, double > rpr)

  Update table with strobes/PID calculation results
- public double PID (double disbalance)

  Calculating the phase correction from the disbalance on strobes Ys
- public void SaveConfigFile ()

  Save Config file in Config directory of Axel-hub
- public void OpenConfigFile ()

  Open Config file from Config directory of Axel-hub

Variables

- public struct Axel_hub::FringeParams realSampling

  Interaction logic for UserControl1.xaml
- private string ArrangedPartner = ""
- TimeSpan totalTime
- TimeSpan currentTime
- public DispatcherTimer dTimer
- GeneralOptions genOptions = null

- public ScanModes scanModes = null
- private bool _Running
- public bool Running

  Some visual adjustments when ADC24 starts/stops
- public RemoteMode remoteMode

  Current remode mode - defines the context next group shots
- public event RemoteModeHandler OnRemoteMode
- public RemoteMessaging remote { get
- set
- public event StartHandler OnStart
- public event RemoteHandler OnRemote
- public event LogHandler OnLog
- private bool _jumboButton = true
- private bool jumboButton

  Set the main scan button to Jumbo mode
- public event ActiveRemoteHandler OnActiveRemote
- private MMexec grpMME
- private MMexec lastMMEin
- private MMexec lastMMEout
- private bool _PID_Enabled
- public bool PID_Enabled

  PID follow the strobe position
- private double lastContrast = -1
- int runI = 0
- string configFile
- List< double > iStack
- List< double > dStack
- private FileLogger logger
- public Point Down
- public Point Up
- public Point Low
- string[ ] Titles = { "runI", "tP", "tI", "tD", "Down.X", "Up.X", "disbal", "corr", "iSD-R", "contrast" }

6.1.1 Function Documentation

6.1.1.1 ActiveRemote()

```
protected void Axel_hub::ActiveRemote (
            bool active )
```

Definition at line 460 of file scanUC.xaml.cs.

6.1.1.2 backMME()

```
public MMexec Axel_hub::backMME (
            int runID,
            double asymmetry,
            MMexec mme = null )
```

Prepare back message with new Raman phase value

Parameters

runID	Shot number
asymmetry	Asymmetry
mme	mme is ONLY for incoming axel-probe feed

Returns

lastMMEout.mmexec.Equals("");

Definition at line 368 of file strobesUC.xaml.cs.

6.1.1.3 calcContrast()

```
public double Axel_hub::calcContrast (
            double A )
```

Calculating contrast

Parameters

A	Asymetry (signal)

Returns

Calculated contrast

Definition at line 251 of file strobesUC.xaml.cs.

6.1.1.4 centreFringe()

```
public double Axel_hub::centreFringe ( )
```

Calculating fringe centre

Returns

Definition at line 241 of file strobesUC.xaml.cs.

6.1.1.5 deconstructAccel()

```
public Dictionary<string, double> Axel_hub::deconstructAccel (
            double accel,
            double mems )
```

Deconstructing accaleration to acceleration components - see dictionary keys

Parameters

accel	acceleration [mg] - target(real)
mems	mems accel.[mg] - measured (real + noise)

Returns

Definition at line 271 of file strobesUC.xaml.cs.

6.1.1.6 dispatcherTimer_Tick()

```
private void Axel_hub::dispatcherTimer_Tick (
            object sender,
            EventArgs e )
```

Shows visual progress of ADC24 acquisition

Parameters

sender	
e	

Definition at line 145 of file scanUC.xaml.cs.

6.1.1.7 fillReport()

```
private void Axel_hub::fillReport (
            Dictionary< string, double > rpr )
```

Update table with strobes/PID calculation results

Parameters

rpr	

Definition at line 423 of file strobesUC.xaml.cs.

6.1.1.8 Flip()

```
public void Axel_hub::Flip ( )
```

Exchange UP/DOWN strobe positions

Definition at line 178 of file strobesUC.xaml.cs.

6.1.1.9 GetBufferSize()

```
public int Axel_hub::GetBufferSize ( )
```

Get the buffer size depending of settings

Returns

Definition at line 342 of file scanUC.xaml.cs.

6.1.1.10 GetSamplingPeriod()

```
public double Axel_hub::GetSamplingPeriod ( )
```

Get the sampling period regardless the units

Returns

[s]

Definition at line 315 of file scanUC.xaml.cs.

6.1.1.11 Image_MouseDown()

```
private void Axel_hub::Image_MouseDown (
            object sender,
            MouseButtonEventArgs e )
```

Definition at line 305 of file scanUC.xaml.cs.

6.1.1.12 Init()

```
public void Axel_hub::Init (
            string _prefix )
```

Initiate strobe from file settings

Parameters

_prefix	

Definition at line 189 of file strobesUC.xaml.cs.

6.1.1.13 InitOptions()

```
public void Axel_hub::InitOptions (
            ref GeneralOptions _genOptions,
            ref ScanModes _scanModes )
```

Initialize - set genOptions

Parameters

_genOptions	From options windows
_scanModes	From saved last used modes

Definition at line 76 of file scanUC.xaml.cs.

6.1.1.14 LogEvent()

```
public void Axel_hub::LogEvent (
            string txt,
            Color?  clr = null )
```

Definition at line 289 of file scanUC.xaml.cs.

6.1.1.15 LogHandler()

```
public delegate void Axel_hub::LogHandler (
            string txt,
            Color?  clr = null )
```

Log into left text box

Log event for massage export

Parameters

txt	
clr	

6.1.1.16 nextShot()

```
public double Axel_hub::nextShot (
            int runID,
            double asymmetry,
            out double correction )
```

Calculated phaseCorr - corrected Raman phase (0 if not PID)

Parameters

runID	Shot number
asymmetry	Asymetry value
correction	The correction value

Returns

The corrected position

Definition at line 307 of file strobesUC.xaml.cs.

6.1.1.17 OnActiveComm()

```
private void Axel_hub::OnActiveComm (
            bool active,
            bool forced )
```

Event when the connection goes ON/OFF

Parameters

active	

Definition at line 469 of file scanUC.xaml.cs.

6.1.1.18 OnAsyncSend()

```
protected void Axel_hub::OnAsyncSend (
            bool OK,
            string json2send )
```

Report sent message in log

Parameters

OK	
json2send	

Definition at line 299 of file scanUC.xaml.cs.

6.1.1.19 OnJumboRepeat()

```
public void Axel_hub::OnJumboRepeat (
            double _fringeScale,
            double _fringeShift,
            MMexec _grpMME,
            double contrastV )
```

Call this before each Jumbo Repeat for group MMexec and modes synchronization

Parameters

_fringeScale	
_fringeShift	
_grpMME	
contrastV	

Definition at line 203 of file strobesUC.xaml.cs.

6.1.1.20 OnRealSampling()

```
public void Axel_hub::OnRealSampling (
            double _realSampling )
```

Definition at line 134 of file scanUC.xaml.cs.

6.1.1.21 OnReceive()

```
private bool Axel_hub::OnReceive (
            string message )
```

Incomming from MM2/Axel-probe message

Parameters

message	

Returns

Definition at line 242 of file scanUC.xaml.cs.

6.1.1.22 OpenConfigFile()

```
public void Axel_hub::OpenConfigFile ( )
```

Open Config file from Config directory of Axel-hub

Definition at line 516 of file strobesUC.xaml.cs.

6.1.1.23 PID()

```
public double Axel_hub::PID (
            double disbalance )
```

Calculating the phase correction from the disbalance on strobes Ys

Parameters

disbalance	

Returns

Definition at line 457 of file strobesUC.xaml.cs.

6.1.1.24 RemoteEvent()

```
protected void Axel_hub::RemoteEvent (
            string msg )
```

Definition at line 277 of file scanUC.xaml.cs.

6.1.1.25 RemoteHandler()

```
public delegate void Axel_hub::RemoteHandler (
            string msg )
```

Incomming message event thingy

Parameters

msg	

6.1.1.26 RemoteModeEvent()

```
protected void Axel_hub::RemoteModeEvent (
            RemoteMode oldMode,
            RemoteMode newMode )
```

Definition at line 231 of file scanUC.xaml.cs.

6.1.1.27 Reset()

```
public void Axel_hub::Reset ( )
```

Initaile strobe for axel-probe simulated fringes

Definition at line 169 of file strobesUC.xaml.cs.

6.1.1.28 SaveConfigFile()

```
public void Axel_hub::SaveConfigFile ( )
```

Save Config file in Config directory of Axel-hub

Definition at line 502 of file strobesUC.xaml.cs.

6.1.1.29 scanClass()

```
public Axel_hub::scanClass ( )
```

Class constructor - set defaults

Definition at line 55 of file scanUC.xaml.cs.

6.1.1.30 SendJson()

```
public bool Axel_hub::SendJson (
            string json,
            bool async = false )
```

Wrapper of remote.sendCommand

Parameters

json	
async	

Returns

Definition at line 107 of file scanUC.xaml.cs.

6.1.1.31 SetActivity()

```
public void Axel_hub::SetActivity (
            string act )
```

Definition at line 114 of file scanUC.xaml.cs.

6.1.1.32 SetFringeParams()

```
public void Axel_hub::SetFringeParams (
            FringeParams fp )
```

Show fringes params

Parameters

fp	fringes params

Definition at line 129 of file scanUC.xaml.cs.

6.1.1.33 SetSamplingRate()

```
public void Axel_hub::SetSamplingRate (
            int rate )
```

Definition at line 119 of file scanUC.xaml.cs.

6.1.1.34 StartDelegate()

```
public delegate void Axel_hub::StartDelegate ( )
```

6.1.1.35 StartEvent()

```
protected void Axel_hub::StartEvent (
            bool jumbo,
            bool down,
            double period,
            int sizeLimit )
```

Definition at line 266 of file scanUC.xaml.cs.

6.1.1.36 StartHandler()

```
public delegate void Axel_hub::StartHandler (
            bool jumbo,
            bool down,
            double period,
            int sizeLimit )
```

Start/Stop group operation wity ADC24 params

Parameters

jumbo	
down	
period	
sizeLimit	

6.1.1.37 Status()

```
private void Axel_hub::Status (
            string sts )
```

Definition at line 161 of file scanUC.xaml.cs.

6.1.1.38 strobesUC()

```
public Axel_hub::strobesUC ( )
```

Class constructor

Definition at line 157 of file strobesUC.xaml.cs.

6.1.1.39 UpdateModes()

```
public void Axel_hub::UpdateModes ( )
```

Set internal from visual modes

Definition at line 92 of file scanUC.xaml.cs.

6.1.1.40 UserControl_Loaded()

```
private void Axel_hub::UserControl_Loaded (
            object sender,
            RoutedEventArgs e )
```

Some secondary to contructor initialilzations

Parameters

sender	
e	

Definition at line 494 of file scanUC.xaml.cs.

6.1.1.41 zeroFringe()

```
public double Axel_hub::zeroFringe ( )
```

Calculating zeroFringe - similar to centreFring but woth phase shift compensation

Returns

Calculated zeroFringe [rad] [-pi..pi]

Definition at line 260 of file strobesUC.xaml.cs.

6.1.2 Variable Documentation

6.1.2.1 _jumboButton

```
private bool Axel_hub::_jumboButton = true
```

Definition at line 360 of file scanUC.xaml.cs.

6.1.2.2 _PID_Enabled

```
private bool Axel_hub::_PID_Enabled
```

Definition at line 114 of file strobesUC.xaml.cs.

6.1.2.3 _Running

```
private bool Axel_hub::_Running
```

Definition at line 166 of file scanUC.xaml.cs.

6.1.2.4 ArrangedPartner

```
private string Axel_hub::ArrangedPartner = ""
```

Definition at line 47 of file scanUC.xaml.cs.

6.1.2.5 configFile

```
string Axel_hub::configFile
```

Definition at line 146 of file strobesUC.xaml.cs.

6.1.2.6 currentTime

```
TimeSpan Axel_hub::currentTime
```

Definition at line 49 of file scanUC.xaml.cs.

6.1.2.7 Down

```
public Point Axel_hub::Down
```

Definition at line 150 of file strobesUC.xaml.cs.

6.1.2.8 dStack

```
List<double> Axel_hub::dStack
```

Definition at line 147 of file strobesUC.xaml.cs.

6.1.2.9 dTimer

```
public DispatcherTimer Axel_hub::dTimer
```

Definition at line 50 of file scanUC.xaml.cs.

6.1.2.10 genOptions

```
GeneralOptions Axel_hub::genOptions = null
```

Definition at line 68 of file scanUC.xaml.cs.

6.1.2.11 grpMME

```
private MMexec Axel_hub::grpMME
```

Definition at line 112 of file strobesUC.xaml.cs.

6.1.2.12 iStack

```
List<double> Axel_hub::iStack
```

Definition at line 147 of file strobesUC.xaml.cs.

6.1.2.13 jumboButton

```
private bool Axel_hub::jumboButton
```

Set the main scan button to Jumbo mode

Definition at line 365 of file scanUC.xaml.cs.

6.1.2.14 lastContrast

```
private double Axel_hub::lastContrast = -1
```

Definition at line 144 of file strobesUC.xaml.cs.

6.1.2.15 lastMMEin

```
private MMexec Axel_hub::lastMMEin
```

Definition at line 112 of file strobesUC.xaml.cs.

6.1.2.16 lastMMEout

```
private MMexec Axel_hub::lastMMEout
```

Definition at line 112 of file strobesUC.xaml.cs.

6.1.2.17 logger

```
private FileLogger Axel_hub::logger
```

Definition at line 148 of file strobesUC.xaml.cs.

6.1.2.18 Low

```
public Point Axel_hub::Low
```

Definition at line 152 of file strobesUC.xaml.cs.

6.1.2.19 OnActiveRemote

```
public event ActiveRemoteHandler Axel_hub::OnActiveRemote
```

Definition at line 459 of file scanUC.xaml.cs.

6.1.2.20 OnLog

```
public event LogHandler Axel_hub::OnLog
```

Definition at line 288 of file scanUC.xaml.cs.

6.1.2.21 OnRemote

```
public event RemoteHandler Axel_hub::OnRemote
```

Definition at line 276 of file scanUC.xaml.cs.

6.1.2.22 OnRemoteMode

```
public event RemoteModeHandler Axel_hub::OnRemoteMode
```

Definition at line 230 of file scanUC.xaml.cs.

6.1.2.23 OnStart

```
public event StartHandler Axel_hub::OnStart
```

Definition at line 265 of file scanUC.xaml.cs.

6.1.2.24 PID_Enabled

```
public bool Axel_hub::PID_Enabled
```

Initial value:

```
{
            get { return _PID_Enabled; }
            set
            {
                _PID_Enabled = value;
                if (value) lbTitle.Content = "PID - ON";
                else lbTitle.Content = "PID - OFF";
            }
        }
        private bool LogPID { get { return chkPIDlog.IsChecked.Value; } }
        private bool Rpr2file { get { return chkRpr2file.IsChecked.Value; } }
        public string prefix { get; private set; }
        public double kP { get { return ndKP.Value; } private set { ndKP.Value = value; } }
        public double kI { get { return ndKI.Value; } private set { ndKI.Value = value; } }
        public double kD { get { return ndKD.Value; } private set { ndKD.Value = value; } }
        public int PiWeight { get { return ndPiWeight.Value; } private set { ndPiWeight.Value = value; } }
        public int kIdepth { get { return ndKIdepth.Value; } private set { ndKIdepth.Value = value; } }
        public int kDdepth { get { return ndKDdepth.Value; } private set { ndKDdepth.Value = value; } }
        public int FreqContrast { get { return ndFreqContrast.Value; } private set { ndFreqContrast.Value =
       value; } }
        public Dictionary<string, double> accelSet { get; private set; }
        public double fringeScale { get; private set; }
        public double fringeShift { get; private set; }
        public double disbalNorm { get; private set; }
        private double refContrast = -1
```

PID follow the strobe position

Definition at line 119 of file strobesUC.xaml.cs.

6.1.2.25 realSampling

```
public struct Axel_hub::FringeParams Axel_hub::realSampling
```

Interaction logic for UserControl1.xaml

6.1.2.26 remote

```
public RemoteMessaging Axel_hub::remote { get
```

Definition at line 236 of file scanUC.xaml.cs.

6.1.2.27 remoteMode

```
public RemoteMode Axel_hub::remoteMode
```

Initial value:

```
{
            get { return _remoteMode; }
            set
            {
                lbMode.Content = "Oper.Mode: " + value.ToString();
                if (value == RemoteMode.Simple_Repeat || value == RemoteMode.Simple_Scan || value ==
        RemoteMode.Disconnected) bbtnStart.Visibility = System.Windows.Visibility.Collapsed;
                else bbtnStart.Visibility = System.Windows.Visibility.Visible;
                RemoteMode tempRemoteMode = _remoteMode; _remoteMode = value; scanModes.remoteMode = value;
                if (!tempRemoteMode.Equals(value)) RemoteModeEvent(tempRemoteMode, value);
                tabControl.IsEnabled = (value == RemoteMode.Ready_To_Remote || value ==
        RemoteMode.Disconnected);
            }
         }
         public delegate void RemoteModeHandler(RemoteMode oldMode, RemoteMode newMode)
```

Current remode mode - defines the context next group shots

Definition at line 217 of file scanUC.xaml.cs.

6.1.2.28 runI

```
int Axel_hub::runI = 0
```

Definition at line 145 of file strobesUC.xaml.cs.

6.1.2.29 Running

```
public bool Axel_hub::Running
```

Some visual adjustments when ADC24 starts/stops

Definition at line 171 of file scanUC.xaml.cs.

6.1.2.30 scanModes

```
public ScanModes Axel_hub::scanModes = null
```

Definition at line 69 of file scanUC.xaml.cs.

6.1.2.31 set

```
Axel_hub::set
```

Definition at line 236 of file scanUC.xaml.cs.

6.1.2.32 Titles

```
string [] Axel_hub::Titles = { "runI", "tP", "tI", "tD", "Down.X", "Up.X", "disbal", "corr",
"iSD-R", "contrast" }
```

Definition at line 418 of file strobesUC.xaml.cs.

6.1.2.33 totalTime

```
TimeSpan Axel_hub::totalTime
```

Definition at line 49 of file scanUC.xaml.cs.

6.1.2.34 Up

```
public Point Axel_hub::Up
```

Definition at line 151 of file strobesUC.xaml.cs.

6.2 Axel_hub::Properties Namespace Reference

Classes

- class Resources

  A strongly-typed resource class, for looking up localized strings, etc.
- class Settings

6.3 OptionsNS Namespace Reference

Classes

- class GeneralOptions

  general options from Options dialog window accesable everywhere
- class Modes

  Visuals and prameters for Top: Axel-chart Middle: Signal panel charts Bottom: Scan and Accel trend tabs/charts
- class OptionsWindow

  OptionsWindow
- class ScanModes

  visuals for the app, MEMS aqcuisition params and scan modes

Enumerations

- enum RemoteMode {
  RemoteMode.Disconnected, RemoteMode.Jumbo_Scan, RemoteMode.Jumbo_Repeat, RemoteMode.Simple_Scan,
  RemoteMode.Simple_Repeat, RemoteMode.Ready_To_Remote }

  The mode negotiated with MM2

6.3.1 Enumeration Type Documentation

6.3.1.1 RemoteMode

```
enum OptionsNS.RemoteMode  [strong]
```

The mode negotiated with MM2

Enumerator

Disconnected	
Jumbo_Scan	
Jumbo_Repeat	
Simple_Scan	
Simple_Repeat	
Ready_To_Remote	

Definition at line 24 of file OptionsType.cs.

6.4 XamlGeneratedNamespace Namespace Reference

Classes

- class GeneratedInternalTypeHelper

  GeneratedInternalTypeHelper

Chapter 7

Class Documentation

7.1 Axel_hub.accelCalibr Struct Reference

Acceleration calibration with optional temperature compensation particular to each MEMS device

Public Member Functions

- double accel (double accelV, double temperV, bool tempComp=false)

  The actual calibration from [V] to [mg] with optional temperature compensation

Public Attributes

- string model
- string SN
- double rAccel
- double rTemper
- double cK0
- double cK1
- double[ ] pK0
- double[ ] pK1

7.1.1 Detailed Description

Acceleration calibration with optional temperature compensation particular to each MEMS device

Definition at line 21 of file AxelHMems.cs.

7.1.2 Member Function Documentation

7.1.2.1 accel()

```
double Axel_hub.accelCalibr.accel (
            double accelV,
            double temperV,
            bool tempComp = false )
```

The actual calibration from [V] to [mg] with optional temperature compensation

Parameters

accelV	
temperV	
tempComp	

Returns

Definition at line 39 of file AxelHMems.cs.

7.1.3 Member Data Documentation

7.1.3.1 cK0

```
double Axel_hub.accelCalibr.cK0
```

Definition at line 27 of file AxelHMems.cs.

7.1.3.2 cK1

```
double Axel_hub.accelCalibr.cK1
```

Definition at line 28 of file AxelHMems.cs.

7.1.3.3 model

```
string Axel_hub.accelCalibr.model
```

Definition at line 23 of file AxelHMems.cs.

7.1.3.4 pK0

```
double [] Axel_hub.accelCalibr.pK0
```

Definition at line 29 of file AxelHMems.cs.

7.1.3.5 pK1

```
double [] Axel_hub.accelCalibr.pK1
```

Definition at line 30 of file AxelHMems.cs.

7.1.3.6 rAccel

```
double Axel_hub.accelCalibr.rAccel
```

Definition at line 25 of file AxelHMems.cs.

7.1.3.7 rTemper

```
double Axel_hub.accelCalibr.rTemper
```

Definition at line 26 of file AxelHMems.cs.

7.1.3.8 SN

```
string Axel_hub.accelCalibr.SN
```

Definition at line 24 of file AxelHMems.cs.

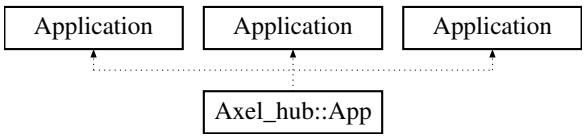
The documentation for this struct was generated from the following file:

- AxelHMems.cs

7.2 Axel_hub::App Class Reference

Interaction logic for App.xaml

Inheritance diagram for Axel_hub::App:



7.2.1 Detailed Description

Interaction logic for App.xaml

App

Definition at line 14 of file App.xaml.cs.

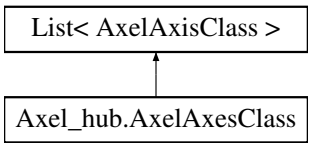
The documentation for this class was generated from the following files:

- App.xaml.cs
- App.g.cs
- App.g.i.cs

7.3 Axel_hub.AxelAxesClass Class Reference

Intermediator between incomming data flow from ucScan user component and AxelAxis user components

Inheritance diagram for Axel_hub.AxelAxesClass:



Public Member Functions

- int prfIdx (string prf)

  Get an index from a prefix (X/Y)
- void Clear (bool Top=true, bool Middle=true, bool Bottom=true)

  Clear and initialize visuals according to the switches
- AxelAxesClass (ref GeneralOptions _genOptions, ref scanClass _ucScan)

  Class constructor
- void AddAxis (ref AxelAxisClass AxelAxis, string prefix)

  The correct way to introduce new axis
- AxelAxisClass byName (char prefix)

  Get an axis by prefix
- void UpdateFromOptions (bool activeComm)

  When the options change, make everybody knows
- delegate void LogHandler (string txt, Color? clr=null)

  The correct way to log a text on the text-box on the left
- void set2startADC24 (bool down, double samplingPeriod, int InnerBufferSize)

  Initialize ADC24 for a new measurement
- void SaveDefaultModes ()

  Save the visual options
- void startADC (bool down, double period, int InnerBufferSize)

  Start new measurement with ADC24
- void DoAcquire (List< Point > dt, out bool next)

Get the acquisition buffer and distribute the data to axes

- void DoAcquireTemperature (List< Point > dt, out bool next)

  Acquire the temperature measurements and send the average to the corresponding axelChart
- void DoRemote (string json)

  The main MOT data getting method format shot.X / shot.Y OR shotData. IMPORTANT for Jumbo-repeat only .X/.Y if .X / .Y runID's are independant for each axis
- void DoJumboScan (bool down)

  When in Jumbo mode Start/Stop the first part of it
- void SetChartStrobes (bool enabled)
- void jumboRepeat (int cycles)

  When in Jumbo mode start the second part of it
- void Closing (object sender, System.ComponentModel.CancelEventArgs e)

  Not destroying anything, just preparing for closing

Public Attributes

- AxelMems axelMems = null

Protected Member Functions

- void LogEvent (string txt, Color? clr=null)

Properties

- int rCount `[get, set]`

  number of real (active) Axes
- bool memsRunning `[get, set]`

  Mask running for the active axelChart

Events

- LogHandler OnLog

7.3.1 Detailed Description

Intermediator between incomming data flow from ucScan user component and AxelAxis user components

- encapsulate not-axis-specific objects (e.g axelMems) and operations (e.g. DoAcquire)

Definition at line 22 of file AxelAxes.cs.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 AxelAxesClass()

```
Axel_hub.AxelAxesClass.AxelAxesClass (
            ref GeneralOptions _genOptions,
            ref scanClass _ucScan )
```

Class constructor

Parameters

_genOptions	general for the app options
_ucScan	the scan user user component ref

Definition at line 98 of file AxelAxes.cs.

7.3.3 Member Function Documentation

7.3.3.1 AddAxis()

```
void Axel_hub.AxelAxesClass.AddAxis (
            ref AxelAxisClass AxelAxis,
            string prefix )
```

The correct way to introduce new axis

Parameters

AxelAxis	
prefix	X or Y

Definition at line 115 of file AxelAxes.cs.

7.3.3.2 byName()

```
AxelAxisClass Axel_hub.AxelAxesClass.byName (
            char prefix )
```

Get an axis by prefix

Parameters

prefix	

Returns

Definition at line 128 of file AxelAxes.cs.

7.3.3.3 Clear()

```
void Axel_hub.AxelAxesClass.Clear (
            bool Top = true,
            bool Middle = true,
            bool Bottom = true )
```

Clear and initialize visuals according to the switches

Parameters

Top	top panel
Middle	middle panel
Bottom	bottom panel

Definition at line 86 of file AxelAxes.cs.

7.3.3.4 Closing()

```
void Axel_hub.AxelAxesClass.Closing (
            object sender,
            System.ComponentModel.CancelEventArgs e )
```

Not destroying anything, just preparing for closing

Parameters

sender	
e	

Definition at line 618 of file AxelAxes.cs.

7.3.3.5 DoAcquire()

```
void Axel_hub.AxelAxesClass.DoAcquire (
            List< Point > dt,
            out bool next )
```

Get the acquisition buffer and distribute the data to axes

Parameters

dt	
next	

Definition at line 246 of file AxelAxes.cs.

7.3.3.6 DoAcquireTemperature()

```
void Axel_hub.AxelAxesClass.DoAcquireTemperature (
            List< Point > dt,
            out bool next )
```

Acquire the temperature measurements and send the average to the corresponding axelChart

Parameters

dt	
next	

Definition at line 296 of file AxelAxes.cs.

7.3.3.7 DoJumboScan()

```
void Axel_hub.AxelAxesClass.DoJumboScan (
            bool down )
```

When in Jumbo mode Start/Stop the first part of it

Parameters

down	

Definition at line 473 of file AxelAxes.cs.

7.3.3.8 DoRemote()

```
void Axel_hub.AxelAxesClass.DoRemote (
            string json )
```

The main MOT data getting method format shot.X / shot.Y OR shotData. IMPORTANT for Jumbo-repeat only .X/.Y if .X / .Y runID's are independant for each axis

Parameters

json	Whatever is comming, it must be formatted according to "Book of JaSON"

Definition at line 351 of file AxelAxes.cs.

7.3.3.9 jumboRepeat()

```
void Axel_hub.AxelAxesClass.jumboRepeat (
            int cycles )
```

When in Jumbo mode start the second part of it

Parameters

cycles	set the number of shots or -1 to continues measurement

Definition at line 541 of file AxelAxes.cs.

7.3.3.10 LogEvent()

```
void Axel_hub.AxelAxesClass.LogEvent (
            string txt,
            Color?  clr = null )  [protected]
```

Definition at line 161 of file AxelAxes.cs.

7.3.3.11 LogHandler()

```
delegate void Axel_hub.AxelAxesClass.LogHandler (
            string txt,
            Color?  clr = null )
```

The correct way to log a text on the text-box on the left

Parameters

txt	
clr	

7.3.3.12 prfIdx()

```
int Axel_hub.AxelAxesClass.prfIdx (
            string prf )
```

Get an index from a prefix (X/Y)

Parameters

prf	

Returns

Definition at line 53 of file AxelAxes.cs.

7.3.3.13 SaveDefaultModes()

```
void Axel_hub.AxelAxesClass.SaveDefaultModes ( )
```

Save the visual options

Definition at line 183 of file AxelAxes.cs.

7.3.3.14 set2startADC24()

```
void Axel_hub.AxelAxesClass.set2startADC24 (
            bool down,
            double samplingPeriod,
            int InnerBufferSize )
```

Initialize ADC24 for a new measurement

Parameters

down	
samplingPeriod	
InnerBufferSize	

Definition at line 172 of file AxelAxes.cs.

7.3.3.15 SetChartStrobes()

```
void Axel_hub.AxelAxesClass.SetChartStrobes (
            bool enabled )
```

Definition at line 531 of file AxelAxes.cs.

7.3.3.16 startADC()

```
void Axel_hub.AxelAxesClass.startADC (
            bool down,
            double period,
            int InnerBufferSize )
```

Start new measurement with ADC24

Parameters

down	
period	
InnerBufferSize	

Definition at line 195 of file AxelAxes.cs.

7.3.3.17 UpdateFromOptions()

```
void Axel_hub.AxelAxesClass.UpdateFromOptions (
            bool activeComm )
```

When the options change, make everybody knows

Parameters

activeComm	

Definition at line 145 of file AxelAxes.cs.

7.3.4 Member Data Documentation

7.3.4.1 axelMems

```
AxelMems Axel_hub.AxelAxesClass.axelMems = null
```

Definition at line 24 of file AxelAxes.cs.

7.3.5 Property Documentation

7.3.5.1 memsRunning

```
bool Axel_hub.AxelAxesClass.memsRunning  [get], [set]
```

Mask running for the active axelChart

Definition at line 67 of file AxelAxes.cs.

7.3.5.2 rCount

```
int Axel_hub.AxelAxesClass.rCount  [get], [set]
```

number of real (active) Axes

Definition at line 35 of file AxelAxes.cs.

7.3.6 Event Documentation

7.3.6.1 OnLog

```
LogHandler Axel_hub.AxelAxesClass.OnLog
```

Definition at line 160 of file AxelAxes.cs.

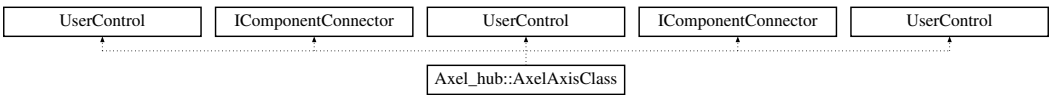
The documentation for this class was generated from the following file:

- AxelAxes.cs

7.4 Axel_hub::AxelAxisClass Class Reference

Interaction logic for AxelAxisUC.xaml AxelAxisClass repressents a single axis of acceleration encapsulated and accesable in AxelAxes list of AxelAxisClass Future intermediator abstract movement (linear or rotation) component will be implemented.

Inheritance diagram for Axel_hub::AxelAxisClass:



7.4.1 Detailed Description

Interaction logic for AxelAxisUC.xaml AxelAxisClass repressents a single axis of acceleration encapsulated and accesable in AxelAxes list of AxelAxisClass Future intermediator abstract movement (linear or rotation) component will be implemented.

AxelAxisClass

Definition at line 49 of file AxelAxisUC.xaml.cs.

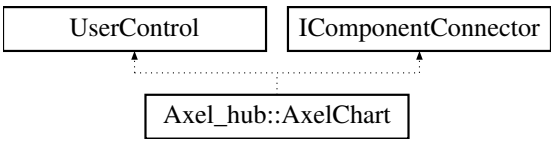
The documentation for this class was generated from the following files:

- AxelAxisUC.xaml.cs
- AxelAxisUC.g.cs
- AxelAxisUC.g.i.cs

7.5 Axel_hub::AxelChart Class Reference

AxelChart

Inheritance diagram for Axel_hub::AxelChart:



7.5.1 Detailed Description

AxelChart

Definition at line 42 of file AxelChart.g.i.cs.

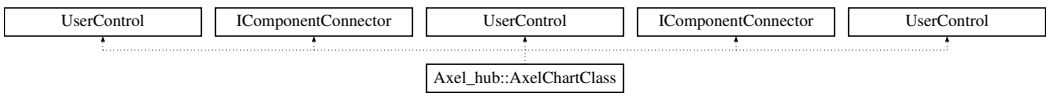
The documentation for this class was generated from the following file:

- AxelChart.g.i.cs

7.6 Axel_hub::AxelChartClass Class Reference

Interaction logic for AxelChart.xaml

Inheritance diagram for Axel_hub::AxelChartClass:



7.6.1 Detailed Description

Interaction logic for AxelChart.xaml

AxelChartClass

Definition at line 48 of file AxelChartUC.xaml.cs.

The documentation for this class was generated from the following files:

- AxelChartUC.xaml.cs
- AxelChartUC.g.cs
- AxelChartUC.g.i.cs

7.7 Axel_hub.AxelMems Class Reference

The hardware abstraction for MEMS with ADC24 (NI9251) device

Public Types

- enum TimingModes { TimingModes.byNone, TimingModes.byADCtimer, TimingModes.byStopwatch, TimingModes.byBoth }

Public Member Functions

- AxelMems (string hwFile="", string memsFile="")

  Class contructor
- void StartStopwatch ()

  Stopwatch routines
- void SetStopwatch (Stopwatch ext_sw)
- double TimeElapsed ()
- double RealConvRate (double wantedCR)

  Find nearest up sampling freq
- delegate void RealSamplingHandler (double realSampling)
- double[,] readBurst (int nPoints)

  Inner methods of continious (no gaps) data acquisition refer. NI9251 and related documentation
- void configureVITask (string physicalChn, int numbSamples, double samplingRate)
- bool isDevicePlugged ()

  Check for device presence
- void Reset ()

  Reset before new series of measurements
- delegate void AcquireHandler (List< Point > data, out bool next)
- void StartAcquisition (int samplesPerChannel, double samplingRate)

  Set conditions for new data acquisition series
- void StopAcquisition ()

Public Attributes

- bool AdjustTimelineToStopwatch = false

  false - use the set time interval between points true - adjust the time interval to stopwatch markers
- readonly double[ ] FixConvRate

  NI9251 support fixed sampling freq listed here
- TimingModes TimingMode = TimingModes.byNone
- accelCalibr memsX
- Dictionary< string, string > hw = new Dictionary<string, string>()
- int Timeout = -1
- List< double > rawData = null

Protected Member Functions

- void RealSamplingEvent (double realSampling)
- void AcquireEvent (List< Point > data, out bool next)

Properties

- int nSamples `[get]`
- double sampleRate `[get]`
- bool running `[get]`
- int activeChannel `[get, set]`

Events

- RealSamplingHandler OnRealSampling
- AcquireHandler OnAcquire

7.7.1 Detailed Description

The hardware abstraction for MEMS with ADC24 (NI9251) device

Definition at line 65 of file AxelHMems.cs.

7.7.2 Member Enumeration Documentation

7.7.2.1 TimingModes

`enum Axel_hub.AxelMems.TimingModes [strong]`

Enumerator

byNone	
byADCtimer	
byStopwatch	
byBoth	

Definition at line 80 of file AxelHMems.cs.

7.7.3 Constructor & Destructor Documentation

7.7.3.1 AxelMems()

```
Axel_hub.AxelMems.AxelMems (
            string hwFile = "",
            string memsFile = "" )
```

Class contructor

Parameters

hwFile	Hardware file (NI9251 settings)
memsFile	Mems calibration and teperature compensation

Definition at line 117 of file AxelHMems.cs.

7.7.4 Member Function Documentation

7.7.4.1 AcquireEvent()

```
void Axel_hub.AxelMems.AcquireEvent (
            List< Point > data,
            out bool next )  [protected]
```

Definition at line 291 of file AxelHMems.cs.

7.7.4.2 AcquireHandler()

```
delegate void Axel_hub.AxelMems.AcquireHandler (
            List< Point > data,
            out bool next )
```

7.7.4.3 configureVITask()

```
void Axel_hub.AxelMems.configureVITask (
            string physicalChn,
            int numbSamples,
            double samplingRate )
```

Definition at line 224 of file AxelHMems.cs.

7.7.4.4 isDevicePlugged()

```
bool Axel_hub.AxelMems.isDevicePlugged ( )
```

Check for device presence

Returns

Definition at line 244 of file AxelHMems.cs.

7.7.4.5 readBurst()

```
double [,] Axel_hub.AxelMems.readBurst (
            int nPoints )
```

Inner methods of continious (no gaps) data acquisition refer. NI9251 and related documentation

Definition at line 212 of file AxelHMems.cs.

7.7.4.6 RealConvRate()

```
double Axel_hub.AxelMems.RealConvRate (
            double wantedCR )
```

Find nearest up sampling freq

Parameters

wantedCR	Desired freq

Returns

Definition at line 181 of file AxelHMems.cs.

7.7.4.7 RealSamplingEvent()

```
void Axel_hub.AxelMems.RealSamplingEvent (
            double realSampling )  [protected]
```

Definition at line 202 of file AxelHMems.cs.

7.7.4.8 RealSamplingHandler()

```
delegate void Axel_hub.AxelMems.RealSamplingHandler (
            double realSampling )
```

7.7.4.9 Reset()

```
void Axel_hub.AxelMems.Reset ( )
```

Reset before new series of measurements

Definition at line 263 of file AxelHMems.cs.

7.7.4.10 SetStopwatch()

```
void Axel_hub.AxelMems.SetStopwatch (
            Stopwatch ext_sw )
```

Definition at line 157 of file AxelHMems.cs.

7.7.4.11 StartAcquisition()

```
void Axel_hub.AxelMems.StartAcquisition (
            int samplesPerChannel,
            double samplingRate )
```

Set conditions for new data acquisition series

Parameters

samplesPerChannel	
samplingRate	

Definition at line 302 of file AxelHMems.cs.

7.7.4.12 StartStopwatch()

```
void Axel_hub.AxelMems.StartStopwatch ( )
```

Stopwatch routines

Definition at line 152 of file AxelHMems.cs.

7.7.4.13 StopAcquisition()

```
void Axel_hub.AxelMems.StopAcquisition ( )
```

Definition at line 448 of file AxelHMems.cs.

7.7.4.14 TimeElapsed()

```
double Axel_hub.AxelMems.TimeElapsed ( )
```

Definition at line 161 of file AxelHMems.cs.

7.7.5 Member Data Documentation

7.7.5.1 AdjustTimelineToStopwatch

```
bool Axel_hub.AxelMems.AdjustTimelineToStopwatch = false
```

false - use the set time interval between points true - adjust the time interval to stopwatch markers

Definition at line 72 of file AxelHMems.cs.

7.7.5.2 FixConvRate

```
readonly double [] Axel_hub.AxelMems.FixConvRate
```

Initial value:

```
= { 102400, 51200, 34133, 25600, 20480, 17067, 14629, 12800, 11378,
            10240, 9309, 8533, 7314, 6400, 5689, 5120, 4655, 4267, 3657, 3200, 2844, 2560, 2327, 2133, 1829,
            1600, 1422, 1280, 1164, 1067, 914, 800, 711, 640, 582, 533, 457, 400, 356, 320, 291, 267 }
```

NI9251 support fixed sampling freq listed here

Definition at line 76 of file AxelHMems.cs.

7.7.5.3 hw

```
Dictionary<string, string> Axel_hub.AxelMems.hw = new Dictionary<string, string>()
```

Definition at line 90 of file AxelHMems.cs.

7.7.5.4 memsX

```
accelCalibr Axel_hub.AxelMems.memsX
```

Definition at line 88 of file AxelHMems.cs.

7.7.5.5 rawData

```
List<double> Axel_hub.AxelMems.rawData = null
```

Definition at line 95 of file AxelHMems.cs.

7.7.5.6 Timeout

```
int Axel_hub.AxelMems.Timeout = -1
```

Definition at line 94 of file AxelHMems.cs.

7.7.5.7 TimingMode

```
TimingModes Axel_hub.AxelMems.TimingMode = TimingModes.byNone
```

Definition at line 81 of file AxelHMems.cs.

7.7.6 Property Documentation

7.7.6.1 activeChannel

```
int Axel_hub.AxelMems.activeChannel  [get], [set]
```

Definition at line 174 of file AxelHMems.cs.

7.7.6.2 nSamples

```
int Axel_hub.AxelMems.nSamples  [get]
```

Definition at line 92 of file AxelHMems.cs.

7.7.6.3 running

```
bool Axel_hub.AxelMems.running  [get]
```

Definition at line 170 of file AxelHMems.cs.

7.7.6.4 sampleRate

```
double Axel_hub.AxelMems.sampleRate  [get]
```

Definition at line 93 of file AxelHMems.cs.

7.7.7 Event Documentation

7.7.7.1 OnAcquire

```
AcquireHandler Axel_hub.AxelMems.OnAcquire
```

Definition at line 289 of file AxelHMems.cs.

7.7.7.2 OnRealSampling

```
RealSamplingHandler Axel_hub.AxelMems.OnRealSampling
```

Definition at line 200 of file AxelHMems.cs.

The documentation for this class was generated from the following file:

- AxelHMems.cs

7.8 Axel_hub.AxelMemsTemperature Class Reference

The temperature in a class abstraction

Public Member Functions

- AxelMemsTemperature (string hwFile="")
- double[ ] TakeTheTemperature ()

  The actual temperature measurement

Public Attributes

- Dictionary< string, string > hw = new Dictionary<string, string>()

7.8.1 Detailed Description

The temperature in a class abstraction

Definition at line 458 of file AxelHMems.cs.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 AxelMemsTemperature()

```
Axel_hub.AxelMemsTemperature.AxelMemsTemperature (
            string hwFile = "" )
```

Definition at line 462 of file AxelHMems.cs.

7.8.3 Member Function Documentation

7.8.3.1 TakeTheTemperature()

```
double [] Axel_hub.AxelMemsTemperature.TakeTheTemperature ( )
```

The actual temperature measurement

Returns

Definition at line 478 of file AxelHMems.cs.

7.8.4 Member Data Documentation

7.8.4.1 hw

```
Dictionary<string, string> Axel_hub.AxelMemsTemperature.hw = new Dictionary<string, string>()
```

Definition at line 460 of file AxelHMems.cs.

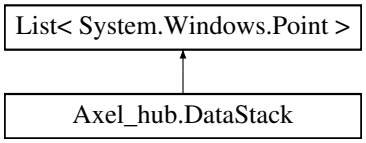
The documentation for this class was generated from the following file:

- AxelHMems.cs

7.9 Axel_hub.DataStack Class Reference

You (developer) need to set TimeMode and one of SizeLimit or TimeLimit TimeMode is about the way DataStack limits its size The output is from standart List method ToArray in order to set DataSource of Graph

Inheritance diagram for Axel_hub.DataStack:



Public Member Functions

- DataStack (int depth=1000, string _prefix="")

  Class constructor
- delegate void RefreshHandler ()
- int Fit2Limit ()

  Restrict the size to about Depth length
- new void Clear ()

  Clean everything up
- new int Add (Point pnt)

  Overriding method to the base, assuming correct time order
- int AddPoint (double Y, double X=double.NaN)

  Add point by coordinates assuming correct time order
- int AddRange (List< Point > pnts)

  Add list of points assuming correct time order
- DataStack CopyEach (int each)

  Copy subset of point (skiping some) to speed up visualization
- DataStack Clone (double offsetX=0, double offsetY=0)

  Clone datastack with some offset applied
- DataStack TimePortion (double fromTime, double toTime)

  Extract sub-DataStack for a time range
- DataStack Portion (int lastNPoints, int backFrom=-1)

  Extract sub-DataStack for an index range
- DataStack Compress (int degree=5)

  Another method (moving average) to extract subset of datastack for speed up visualization
- double[,] ExportToArray ()

  Export data in array[,] format for NI library input
- bool ImportFromArray (double[,] da)

  Import data from array[,] (NI routines)
- int indexByX (double X, bool smart=true)

  Get index by time in time series
- bool statsByIdx (int FromIdx, int ToIdx, bool weightMean, out double Mean, out double stDev)

  Statistics in an index range with averaging method
- bool statsByTime (double endOfTimeInterval, double duration, bool weightMean, out double Mean, out double stDev)

  Statistics in a time range with averaging method
- double[ ] pointXs ()

  Array of X coordinates
- void Rescale (double[ ] newXs, double offsetX=0)

  Change the x scale with new one and offset
- double[ ] pointYs ()

  Array of Y coordinates
- Point pointSDev (bool relativeY=false)

  StandardDeviation by X and Y
- void importFromArrays (double[ ] xs, double[ ] ys)

  Another way to import -> double[] and double[]
- void fillSamples (int n)

  Fill with some random point, mostly for simulation
- bool OpenPair (string fn, ref GroupBox header, int rm=1)

  Open tab separated x,y text file
- void SavePair (string fn, string rem="", string format="")

  Save tab separated x,y text file

Public Attributes

- Dictionary< string, double > RefFileStats
- int visualCountLimit = -1
- int generalIdx = 0
- FileLogger logger
- Stopwatch stopWatch

Static Public Attributes

- const int maxDepth = 15000000

Protected Member Functions

- void RefreshEvent ()

Properties

- string prefix `[get]`
- string rem `[get, set]`
- string lastError `[get, set]`
- bool StackMode `[get, set]`
- bool Running `[get, set]`

  Running the stopwatch and status
- int Depth `[get, set]`
- bool TimeSeriesMode `[get, set]`
- Point First `[get]`

  First data point
- Point Last `[get]`

  Last data point

Events

- RefreshHandler OnRefresh

7.9.1 Detailed Description

You (developer) need to set TimeMode and one of SizeLimit or TimeLimit TimeMode is about the way DataStack limits its size The output is from standart List method ToArray in order to set DataSource of Graph

Definition at line 38 of file DataStackLib.cs.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 DataStack()

```
Axel_hub.DataStack.DataStack (
            int depth = 1000,
            string _prefix = "" )
```

Class constructor

Parameters

depth	-1 for Non-Stack modes
_prefix	

Definition at line 45 of file DataStackLib.cs.

7.9.3 Member Function Documentation

7.9.3.1 Add()

```
new int Axel_hub.DataStack.Add (
            Point pnt )
```

Overriding method to the base, assuming correct time order

Parameters

pnt	

Returns

Definition at line 152 of file DataStackLib.cs.

7.9.3.2 AddPoint()

```
int Axel_hub.DataStack.AddPoint (
            double Y,
            double X = double.NaN )
```

Add point by coordinates assuming correct time order

Parameters

Y	
X	

Returns

Definition at line 166 of file DataStackLib.cs.

7.9.3.3 AddRange()

```
int Axel_hub.DataStack.AddRange (
            List< Point > pnts )
```

Add list of points assuming correct time order

Parameters

pnts	

Returns

Definition at line 182 of file DataStackLib.cs.

7.9.3.4 Clear()

```
new void Axel_hub.DataStack.Clear ( )
```

Clean everything up

Definition at line 119 of file DataStackLib.cs.

7.9.3.5 Clone()

```
DataStack Axel_hub.DataStack.Clone (
            double offsetX = 0,
            double offsetY = 0 )
```

Clone datastack with some offset applied

Parameters

offsetX	
offsetY	

Returns

Definition at line 220 of file DataStackLib.cs.

7.9.3.6 Compress()

```
DataStack Axel_hub.DataStack.Compress (
            int degree = 5 )
```

Another method (moving average) to extract subset of datastack for speed up visualization

Parameters

degree	

Returns

Definition at line 288 of file DataStackLib.cs.

7.9.3.7 CopyEach()

```
DataStack Axel_hub.DataStack.CopyEach (
            int each )
```

Copy subset of point (skiping some) to speed up visualization

Parameters

each	

Returns

Definition at line 198 of file DataStackLib.cs.

7.9.3.8 ExportToArray()

```
double [,] Axel_hub.DataStack.ExportToArray ( )
```

Export data in array[,] format for NI library input

Returns

Definition at line 314 of file DataStackLib.cs.

7.9.3.9 fillSamples()

```
void Axel_hub.DataStack.fillSamples (
            int n )
```

Fill with some random point, mostly for simulation

Parameters

n	

Definition at line 567 of file DataStackLib.cs.

7.9.3.10 Fit2Limit()

```
int Axel_hub.DataStack.Fit2Limit ( )
```

Restrict the size to about Depth length

Returns

Definition at line 108 of file DataStackLib.cs.

7.9.3.11 ImportFromArray()

```
bool Axel_hub.DataStack.ImportFromArray (
            double da[,] )
```

Import data from array[,] (NI routines)

Parameters

da	

Returns

Definition at line 329 of file DataStackLib.cs.

7.9.3.12 importFromArrays()

```
void Axel_hub.DataStack.importFromArrays (
            double[] xs,
            double[] ys )
```

Another way to import -> double[] and double[]

Parameters

xs	
ys	

Definition at line 549 of file DataStackLib.cs.

7.9.3.13 indexByX()

```
int Axel_hub.DataStack.indexByX (
            double X,
            bool smart = true )
```

Get index by time in time series

Parameters

X	
smart	more direct way with equidistance asumption

Returns

Definition at line 363 of file DataStackLib.cs.

7.9.3.14 OpenPair()

```
bool Axel_hub.DataStack.OpenPair (
            string fn,
```

```
            ref GroupBox header,
            int rm = 1 )
```

Open tab separated x,y text file

Parameters

fn	
header	
rm	

Returns

Definition at line 581 of file DataStackLib.cs.

7.9.3.15 pointSDev()

```
Point Axel_hub.DataStack.pointSDev (
            bool relativeY = false )
```

StandardDeviation by X and Y

Parameters

relativeY	

Returns

Definition at line 533 of file DataStackLib.cs.

7.9.3.16 pointXs()

```
double [] Axel_hub.DataStack.pointXs ( )
```

Array of X coordinates

Returns

Definition at line 488 of file DataStackLib.cs.

7.9.3.17 pointYs()

```
double [] Axel_hub.DataStack.pointYs ( )
```

Array of Y coordinates

Returns

Definition at line 518 of file DataStackLib.cs.

7.9.3.18 Portion()

```
DataStack Axel_hub.DataStack.Portion (
            int lastNPoints,
            int backFrom = -1 )
```

Extract sub-DataStack for an index range

Parameters

lastNPoints	
backFrom	

Returns

Definition at line 266 of file DataStackLib.cs.

7.9.3.19 RefreshEvent()

```
void Axel_hub.DataStack.RefreshEvent ( )  [protected]
```

Definition at line 80 of file DataStackLib.cs.

7.9.3.20 RefreshHandler()

```
delegate void Axel_hub.DataStack.RefreshHandler ( )
```

7.9.3.21 Rescale()

```
void Axel_hub.DataStack.Rescale (
            double[] newXs,
            double offsetX = 0 )
```

Change the x scale with new one and offset

Parameters

newXs	
offsetX	

Definition at line 502 of file DataStackLib.cs.

7.9.3.22 SavePair()

```
void Axel_hub.DataStack.SavePair (
            string fn,
            string rem = "",
            string format = "" )
```

Save tab separated x,y text file

Parameters

fn	
rem	
format	

Definition at line 637 of file DataStackLib.cs.

7.9.3.23 statsByIdx()

```
bool Axel_hub.DataStack.statsByIdx (
            int FromIdx,
            int ToIdx,
            bool weightMean,
            out double Mean,
            out double stDev )
```

Statistics in an index range with averaging method

Parameters

FromIdx	
ToIdx	
weightMean	averaging method
Mean	
stDev	

Returns

Definition at line 395 of file DataStackLib.cs.

7.9.3.24 statsByTime()

```
bool Axel_hub.DataStack.statsByTime (
            double endOfTimeInterval,
            double duration,
            bool weightMean,
            out double Mean,
            out double stDev )
```

Statistics in a time range with averaging method

Parameters

endOfTimeInterval	
duration	
weightMean	averaging method
Mean	
stDev	

Returns

Definition at line 439 of file DataStackLib.cs.

7.9.3.25 TimePortion()

```
DataStack Axel_hub.DataStack.TimePortion (
            double fromTime,
            double toTime )
```

Extract sub-DataStack for a time range

Parameters

fromTime	
toTime	

Returns

Definition at line 239 of file DataStackLib.cs.

7.9.4 Member Data Documentation

7.9.4.1 generalIdx

```
int Axel_hub.DataStack.generalIdx = 0
```

Definition at line 65 of file DataStackLib.cs.

7.9.4.2 logger

```
FileLogger Axel_hub.DataStack.logger
```

Definition at line 74 of file DataStackLib.cs.

7.9.4.3 maxDepth

```
const int Axel_hub.DataStack.maxDepth = 15000000  [static]
```

Definition at line 99 of file DataStackLib.cs.

7.9.4.4 RefFileStats

```
Dictionary<string, double> Axel_hub.DataStack.RefFileStats
```

Definition at line 62 of file DataStackLib.cs.

7.9.4.5 stopWatch

```
Stopwatch Axel_hub.DataStack.stopWatch
```

Definition at line 75 of file DataStackLib.cs.

7.9.4.6 visualCountLimit

```
int Axel_hub.DataStack.visualCountLimit = -1
```

Definition at line 64 of file DataStackLib.cs.

7.9.5 Property Documentation

7.9.5.1 Depth

```
int Axel_hub.DataStack.Depth  [get], [set]
```

Definition at line 100 of file DataStackLib.cs.

7.9.5.2 First

```
Point Axel_hub.DataStack.First  [get]
```

First data point

Definition at line 346 of file DataStackLib.cs.

7.9.5.3 Last

```
Point Axel_hub.DataStack.Last  [get]
```

Last data point

Definition at line 353 of file DataStackLib.cs.

7.9.5.4 lastError

```
string Axel_hub.DataStack.lastError  [get], [set]
```

Definition at line 61 of file DataStackLib.cs.

7.9.5.5 prefix

```
string Axel_hub.DataStack.prefix  [get]
```

Definition at line 59 of file DataStackLib.cs.

7.9.5.6 rem

```
string Axel_hub.DataStack.rem  [get], [set]
```

Definition at line 60 of file DataStackLib.cs.

7.9.5.7 Running

```
bool Axel_hub.DataStack.Running  [get], [set]
```

Running the stopwatch and status

Definition at line 90 of file DataStackLib.cs.

7.9.5.8 StackMode

```
bool Axel_hub.DataStack.StackMode  [get], [set]
```

Definition at line 69 of file DataStackLib.cs.

7.9.5.9 TimeSeriesMode

```
bool Axel_hub.DataStack.TimeSeriesMode  [get], [set]
```

Definition at line 102 of file DataStackLib.cs.

7.9.6 Event Documentation

7.9.6.1 OnRefresh

```
RefreshHandler Axel_hub.DataStack.OnRefresh
```

Definition at line 78 of file DataStackLib.cs.

The documentation for this class was generated from the following file:

- DataStackLib.cs

7.10 Axel_hub::FringeParams Struct Reference

fringes(phi) = cos(period ∗ phi + phase) + offset

Public Attributes

- public double period
  in mg per rad
- public double phase
  the MEMS and the interferometer are not entirely paralel
- public double offset
  phase offset [rad]

7.10.1 Detailed Description

fringes(phi) = cos(period ∗ phi + phase) + offset

Definition at line 26 of file scanUC.xaml.cs.

7.10.2 Member Data Documentation

7.10.2.1 offset

```
public double Axel_hub::FringeParams::offset
```

phase offset [rad]

Definition at line 39 of file scanUC.xaml.cs.

7.10.2.2 period

```
public double Axel_hub::FringeParams::period
```

in mg per rad

Definition at line 31 of file scanUC.xaml.cs.

7.10.2.3 phase

```
public double Axel_hub::FringeParams::phase
```

the MEMS and the interferometer are not entirely paralel

Definition at line 35 of file scanUC.xaml.cs.

The documentation for this struct was generated from the following file:

- scanUC.xaml.cs

7.11 OptionsNS.GeneralOptions Class Reference

general options from Options dialog window accesable everywhere

Public Types

- enum SaveModes { SaveModes.save, SaveModes.ask, SaveModes.nosave }

Public Member Functions

- void Save ()

Public Attributes

- SaveModes saveModes

Properties

- int AxesChannels [get, set]
- string SignalCursorPrec [get, set]
- string SignalTablePrec [get, set]
- string SaveFilePrec [get, set]
- string LogFilePrec [get, set]
- bool intN2 [get, set]
- bool saveVisuals [get, set]
- bool followPID [get, set]
- int TrendSignalLen [get, set]
- int RawSignalAvg [get, set]
- bool JumboScan [get, set]
- bool JumboRepeat [get, set]
- bool MemsInJumbo [get, set]
- bool ShowMemsIfRunning [get, set]
- double Mems2SignDelay [get, set]
- double Mems2SignLen [get, set]
- bool TemperatureEnabled [get, set]
- bool TemperatureCompensation [get, set]
- string MemsHw [get, set]
- string TemperatureHw [get, set]

7.11.1 Detailed Description

general options from Options dialog window accesable everywhere

Definition at line 37 of file OptionsType.cs.

7.11.2 Member Enumeration Documentation

7.11.2.1 SaveModes

enum OptionsNS.GeneralOptions.SaveModes [strong]

Enumerator

save	
ask	
nosave	

Definition at line 39 of file OptionsType.cs.

7.11.3 Member Function Documentation

7.11.3.1 Save()

```
void OptionsNS.GeneralOptions.Save ( )
```

Definition at line 74 of file OptionsType.cs.

7.11.4 Member Data Documentation

7.11.4.1 saveModes

```
SaveModes OptionsNS.GeneralOptions.saveModes
```

Definition at line 59 of file OptionsType.cs.

7.11.5 Property Documentation

7.11.5.1 AxesChannels

```
int OptionsNS.GeneralOptions.AxesChannels  [get], [set]
```

Definition at line 42 of file OptionsType.cs.

7.11.5.2 followPID

```
bool OptionsNS.GeneralOptions.followPID  [get], [set]
```

Definition at line 52 of file OptionsType.cs.

7.11.5.3 intN2

```
bool OptionsNS.GeneralOptions.intN2  [get], [set]
```

Definition at line 49 of file OptionsType.cs.

7.11.5.4 JumboRepeat

```
bool OptionsNS.GeneralOptions.JumboRepeat  [get], [set]
```

Definition at line 58 of file OptionsType.cs.

7.11.5.5 JumboScan

```
bool OptionsNS.GeneralOptions.JumboScan  [get], [set]
```

Definition at line 57 of file OptionsType.cs.

7.11.5.6 LogFilePrec

```
string OptionsNS.GeneralOptions.LogFilePrec  [get], [set]
```

Definition at line 47 of file OptionsType.cs.

7.11.5.7 Mems2SignDelay

```
double OptionsNS.GeneralOptions.Mems2SignDelay  [get], [set]
```

Definition at line 65 of file OptionsType.cs.

7.11.5.8 Mems2SignLen

```
double OptionsNS.GeneralOptions.Mems2SignLen  [get], [set]
```

Definition at line 66 of file OptionsType.cs.

7.11.5.9 MemsHw

```
string OptionsNS.GeneralOptions.MemsHw  [get], [set]
```

Definition at line 71 of file OptionsType.cs.

7.11.5.10 MemsInJumbo

```
bool OptionsNS.GeneralOptions.MemsInJumbo  [get], [set]
```

Definition at line 62 of file OptionsType.cs.

7.11.5.11 RawSignalAvg

```
int OptionsNS.GeneralOptions.RawSignalAvg  [get], [set]
```

Definition at line 55 of file OptionsType.cs.

7.11.5.12 SaveFilePrec

```
string OptionsNS.GeneralOptions.SaveFilePrec  [get], [set]
```

Definition at line 46 of file OptionsType.cs.

7.11.5.13 saveVisuals

```
bool OptionsNS.GeneralOptions.saveVisuals  [get], [set]
```

Definition at line 50 of file OptionsType.cs.

7.11.5.14 ShowMemsIfRunning

```
bool OptionsNS.GeneralOptions.ShowMemsIfRunning  [get], [set]
```

Definition at line 63 of file OptionsType.cs.

7.11.5.15 SignalCursorPrec

```
string OptionsNS.GeneralOptions.SignalCursorPrec  [get], [set]
```

Definition at line 44 of file OptionsType.cs.

7.11.5.16 SignalTablePrec

```
string OptionsNS.GeneralOptions.SignalTablePrec  [get], [set]
```

Definition at line 45 of file OptionsType.cs.

7.11.5.17 TemperatureCompensation

```
bool OptionsNS.GeneralOptions.TemperatureCompensation  [get], [set]
```

Definition at line 69 of file OptionsType.cs.

7.11.5.18 TemperatureEnabled

```
bool OptionsNS.GeneralOptions.TemperatureEnabled  [get], [set]
```

Definition at line 68 of file OptionsType.cs.

7.11.5.19 TemperatureHw

```
string OptionsNS.GeneralOptions.TemperatureHw  [get], [set]
```

Definition at line 72 of file OptionsType.cs.

7.11.5.20 TrendSignalLen

```
int OptionsNS.GeneralOptions.TrendSignalLen  [get], [set]
```

Definition at line 54 of file OptionsType.cs.

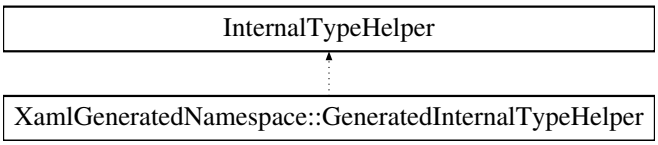
The documentation for this class was generated from the following file:

- OptionsType.cs

7.12 XamlGeneratedNamespace::GeneratedInternalTypeHelper Class Reference

GeneratedInternalTypeHelper

Inheritance diagram for XamlGeneratedNamespace::GeneratedInternalTypeHelper:



7.12.1 Detailed Description

GeneratedInternalTypeHelper

Definition at line 20 of file GeneratedInternalTypeHelper.g.i.cs.

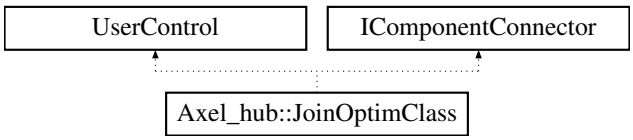
The documentation for this class was generated from the following file:

- GeneratedInternalTypeHelper.g.i.cs

7.13 Axel_hub::JoinOptimClass Class Reference

JoinOptimClass

Inheritance diagram for Axel_hub::JoinOptimClass:



7.13.1 Detailed Description

JoinOptimClass

Definition at line 43 of file JoinOptimUC.g.i.cs.

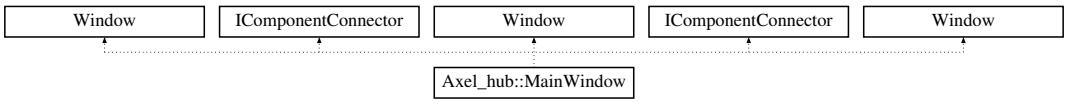
The documentation for this class was generated from the following file:

- JoinOptimUC.g.i.cs

7.14 Axel_hub::MainWindow Class Reference

Interaction logic for MainWindow.xaml command line arguments (space separated): -remote:`partner` -hw↩ :`config.file` where `partner` is remote partner name `title`; hw<c>hardware, `config.file.hw` is in Config folder

Inheritance diagram for Axel_hub::MainWindow:



7.14.1 Detailed Description

Interaction logic for MainWindow.xaml command line arguments (space separated): -remote:`partner` -hw↩ :`config.file` where `partner` is remote partner name `title`; hw<c>hardware, `config.file.hw` is in Config folder

MainWindow

Definition at line 51 of file MainWindow.xaml.cs.

The documentation for this class was generated from the following files:

- MainWindow.xaml.cs
- MainWindow.g.cs
- MainWindow.g.i.cs

7.15 OptionsNS.Modes Class Reference

Visuals and prameters for Top: Axel-chart Middle: Signal panel charts Bottom: Scan and Accel trend tabs/charts

Public Member Functions

- void Save (string prefix)

Properties

- double TopFrame [get, set]
- double TopOfTopFrame [get, set]
- int ShowFreq [get, set]
- int RollMean [get, set]
- int StackDepth [get, set]
- bool ChartUpdate [get, set]
- bool TblUpdate [get, set]
- double PowerCoeff [get, set]
- double MiddleFrame [get, set]
- bool AutoScaleMiddle [get, set]
- bool Background [get, set]
- bool DarkCurrent [get, set]
- bool StdDev [get, set]
- bool N1 [get, set]
- bool N2 [get, set]
- bool RN1 [get, set]
- bool RN2 [get, set]
- bool Ntot [get, set]
- bool RsltTblUpdate [get, set]
- bool RsltChrtUpdate [get, set]
- bool JoinLog [get, set]
- bool SignalLog [get, set]
- double JumboFrom [get, set]
- double JumboTo [get, set]
- double JumboBy [get, set]
- int JumboCycles [get, set]
- bool MemsEnabled [get, set]
- double Kcoeff [get, set]
- double phi0 [get, set]
- double scale [get, set]
- double offset [get, set]
- bool AutoScaleBottom [get, set]
- double kP [get, set]
- double kI [get, set]
- double kD [get, set]
- bool PID_Enabled [get, set]
- bool DoubleStrobe [get, set]

7.15.1 Detailed Description

Visuals and prameters for Top: Axel-chart Middle: Signal panel charts Bottom: Scan and Accel trend tabs/charts

Definition at line 113 of file OptionsType.cs.

7.15.2 Member Function Documentation

7.15.2.1 Save()

```
void OptionsNS.Modes.Save (
            string prefix )
```

Definition at line 160 of file OptionsType.cs.

7.15.3 Property Documentation

7.15.3.1 AutoScaleBottom

```
bool OptionsNS.Modes.AutoScaleBottom  [get], [set]
```

Definition at line 153 of file OptionsType.cs.

7.15.3.2 AutoScaleMiddle

```
bool OptionsNS.Modes.AutoScaleMiddle  [get], [set]
```

Definition at line 127 of file OptionsType.cs.

7.15.3.3 Background

```
bool OptionsNS.Modes.Background  [get], [set]
```

Definition at line 128 of file OptionsType.cs.

7.15.3.4 ChartUpdate

```
bool OptionsNS.Modes.ChartUpdate  [get], [set]
```

Definition at line 121 of file OptionsType.cs.

7.15.3.5 DarkCurrent

```
bool OptionsNS.Modes.DarkCurrent  [get], [set]
```

Definition at line 129 of file OptionsType.cs.

7.15.3.6 DoubleStrobe

```
bool OptionsNS.Modes.DoubleStrobe  [get], [set]
```

Definition at line 158 of file OptionsType.cs.

7.15.3.7 JoinLog

```
bool OptionsNS.Modes.JoinLog  [get], [set]
```

Definition at line 138 of file OptionsType.cs.

7.15.3.8 JumboBy

```
double OptionsNS.Modes.JumboBy  [get], [set]
```

Definition at line 144 of file OptionsType.cs.

7.15.3.9 JumboCycles

```
int OptionsNS.Modes.JumboCycles  [get], [set]
```

Definition at line 145 of file OptionsType.cs.

7.15.3.10 JumboFrom

```
double OptionsNS.Modes.JumboFrom  [get], [set]
```

Definition at line 142 of file OptionsType.cs.

7.15.3.11 JumboTo

```
double OptionsNS.Modes.JumboTo  [get], [set]
```

Definition at line 143 of file OptionsType.cs.

7.15.3.12 Kcoeff

```
double OptionsNS.Modes.Kcoeff  [get], [set]
```

Definition at line 148 of file OptionsType.cs.

7.15.3.13 kD

```
double OptionsNS.Modes.kD  [get], [set]
```

Definition at line 156 of file OptionsType.cs.

7.15.3.14 kI

```
double OptionsNS.Modes.kI  [get], [set]
```

Definition at line 155 of file OptionsType.cs.

7.15.3.15 kP

```
double OptionsNS.Modes.kP  [get], [set]
```

Definition at line 154 of file OptionsType.cs.

7.15.3.16 MemsEnabled

```
bool OptionsNS.Modes.MemsEnabled  [get], [set]
```

Definition at line 147 of file OptionsType.cs.

7.15.3.17 MiddleFrame

```
double OptionsNS.Modes.MiddleFrame  [get], [set]
```

Definition at line 126 of file OptionsType.cs.

7.15.3.18 N1

```
bool OptionsNS.Modes.N1  [get], [set]
```

Definition at line 131 of file OptionsType.cs.

7.15.3.19 N2

```
bool OptionsNS.Modes.N2  [get], [set]
```

Definition at line 132 of file OptionsType.cs.

7.15.3.20 Ntot

```
bool OptionsNS.Modes.Ntot  [get], [set]
```

Definition at line 135 of file OptionsType.cs.

7.15.3.21 offset

```
double OptionsNS.Modes.offset  [get], [set]
```

Definition at line 151 of file OptionsType.cs.

7.15.3.22 phi0

```
double OptionsNS.Modes.phi0  [get], [set]
```

Definition at line 149 of file OptionsType.cs.

7.15.3.23 PID_Enabled

```
bool OptionsNS.Modes.PID_Enabled  [get], [set]
```

Definition at line 157 of file OptionsType.cs.

7.15.3.24 PowerCoeff

```
double OptionsNS.Modes.PowerCoeff  [get], [set]
```

Definition at line 123 of file OptionsType.cs.

7.15.3.25 RN1

```
bool OptionsNS.Modes.RN1  [get], [set]
```

Definition at line 133 of file OptionsType.cs.

7.15.3.26 RN2

```
bool OptionsNS.Modes.RN2  [get], [set]
```

Definition at line 134 of file OptionsType.cs.

7.15.3.27 RollMean

```
int OptionsNS.Modes.RollMean  [get], [set]
```

Definition at line 119 of file OptionsType.cs.

7.15.3.28 RsltChrtUpdate

```
bool OptionsNS.Modes.RsltChrtUpdate  [get], [set]
```

Definition at line 137 of file OptionsType.cs.

7.15.3.29 RsltTblUpdate

```
bool OptionsNS.Modes.RsltTblUpdate  [get], [set]
```

Definition at line 136 of file OptionsType.cs.

7.15.3.30 scale

```
double OptionsNS.Modes.scale  [get], [set]
```

Definition at line 150 of file OptionsType.cs.

7.15.3.31 ShowFreq

```
int OptionsNS.Modes.ShowFreq  [get], [set]
```

Definition at line 118 of file OptionsType.cs.

7.15.3.32 SignalLog

```
bool OptionsNS.Modes.SignalLog  [get], [set]
```

Definition at line 139 of file OptionsType.cs.

7.15.3.33 StackDepth

```
int OptionsNS.Modes.StackDepth  [get], [set]
```

Definition at line 120 of file OptionsType.cs.

7.15.3.34 StdDev

```
bool OptionsNS.Modes.StdDev  [get], [set]
```

Definition at line 130 of file OptionsType.cs.

7.15.3.35 TblUpdate

```
bool OptionsNS.Modes.TblUpdate  [get], [set]
```

Definition at line 122 of file OptionsType.cs.

7.15.3.36 TopFrame

```
double OptionsNS.Modes.TopFrame  [get], [set]
```

Definition at line 116 of file OptionsType.cs.

7.15.3.37 TopOfTopFrame

```
double OptionsNS.Modes.TopOfTopFrame  [get], [set]
```

Definition at line 117 of file OptionsType.cs.

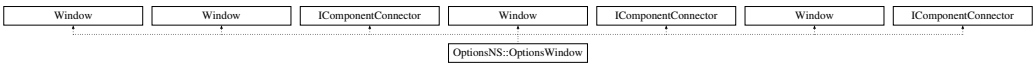
The documentation for this class was generated from the following file:

- OptionsType.cs

7.16 OptionsNS::OptionsWindow Class Reference

OptionsWindow

Inheritance diagram for OptionsNS::OptionsWindow:



7.16.1 Detailed Description

OptionsWindow

Interaction logic, load & save for GeneralOptions genOptions

Definition at line 42 of file Options.g.cs.

The documentation for this class was generated from the following files:

- Options.g.cs
- Options/Options.g.i.cs
- Options.xaml.cs

7.17 Axel_hub::Properties::Resources Class Reference

A strongly-typed resource class, for looking up localized strings, etc.

7.17.1 Detailed Description

A strongly-typed resource class, for looking up localized strings, etc.

Definition at line 25 of file Resources.Designer.cs.

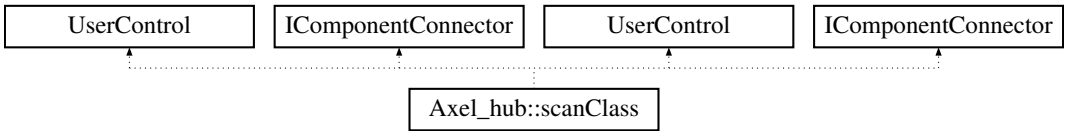
The documentation for this class was generated from the following file:

- Resources.Designer.cs

7.18 Axel_hub::scanClass Class Reference

scanClass

Inheritance diagram for Axel_hub::scanClass:



7.18.1 Detailed Description

scanClass

Definition at line 42 of file scanUC.g.cs.

The documentation for this class was generated from the following files:

- scanUC.g.cs
- scanUC.g.i.cs

7.19 OptionsNS.ScanModes Class Reference

visuals for the app, MEMS aqcuisition params and scan modes

Public Member Functions

- void Save ()

Public Attributes

- RemoteMode remoteMode = RemoteMode.Disconnected

Properties

- double Left [get, set]
- double Top [get, set]
- double Width [get, set]
- double Height [get, set]
- int SamplingFreq [get, set]
- bool TimeLimitMode [get, set]
- int TimeLimit [get, set]
- bool SizeLimitMode [get, set]
- int SizeLimit [get, set]

7.19.1 Detailed Description

visuals for the app, MEMS aqcuisition params and scan modes

Definition at line 83 of file OptionsType.cs.

7.19.2 Member Function Documentation

7.19.2.1 Save()

```
void OptionsNS.ScanModes.Save ( )
```

Definition at line 101 of file OptionsType.cs.

7.19.3 Member Data Documentation

7.19.3.1 remoteMode

```
RemoteMode OptionsNS.ScanModes.remoteMode = RemoteMode.Disconnected
```

Definition at line 99 of file OptionsType.cs.

7.19.4 Property Documentation

7.19.4.1 Height

```
double OptionsNS.ScanModes.Height  [get], [set]
```

Definition at line 89 of file OptionsType.cs.

7.19.4.2 Left

```
double OptionsNS.ScanModes.Left  [get], [set]
```

Definition at line 86 of file OptionsType.cs.

7.19.4.3 SamplingFreq

```
int OptionsNS.ScanModes.SamplingFreq  [get], [set]
```

Definition at line 92 of file OptionsType.cs.

7.19.4.4 SizeLimit

```
int OptionsNS.ScanModes.SizeLimit  [get], [set]
```

Definition at line 96 of file OptionsType.cs.

7.19.4.5 SizeLimitMode

```
bool OptionsNS.ScanModes.SizeLimitMode  [get], [set]
```

Definition at line 95 of file OptionsType.cs.

7.19.4.6 TimeLimit

```
int OptionsNS.ScanModes.TimeLimit  [get], [set]
```

Definition at line 94 of file OptionsType.cs.

7.19.4.7 TimeLimitMode

```
bool OptionsNS.ScanModes.TimeLimitMode  [get], [set]
```

Definition at line 93 of file OptionsType.cs.

7.19.4.8 Top

```
double OptionsNS.ScanModes.Top  [get], [set]
```

Definition at line 87 of file OptionsType.cs.

7.19.4.9 Width

```
double OptionsNS.ScanModes.Width  [get], [set]
```

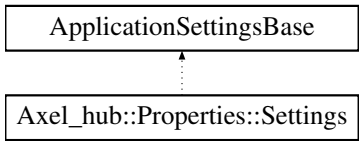
Definition at line 88 of file OptionsType.cs.

The documentation for this class was generated from the following file:

- OptionsType.cs

7.20 Axel_hub::Properties::Settings Class Reference

Inheritance diagram for Axel_hub::Properties::Settings:



7.20.1 Detailed Description

Definition at line 16 of file Settings.Designer.cs.

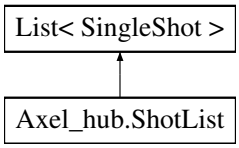
The documentation for this class was generated from the following file:

- Settings.Designer.cs

7.21 Axel_hub.ShotList Class Reference

List / series of single shots

Inheritance diagram for Axel_hub.ShotList:



Public Member Functions

- ShotList (bool arch=true, string FN="", string prefix="")

  Class constructor if arch -> open file if FN not empty, or create FN if empty if not arch -> ignore FN and prefix
- new void Add (SingleShot ss)

  New Add with optional log and size limit
- void resetScan ()

  Reset scan of archive
- SingleShot archiScan (out bool next)

  Get the next scan from a file
- void Save (string FN="")

  Save a file with JSON of single shots per line read the format with ResetScan And ArchiScan

Public Attributes

- Dictionary< string, double > conditions = new Dictionary<string, double>()

Protected Attributes

- int depth = 10000

Properties

- string filename `[get]`
- bool archiveMode `[get]`
- bool savingMode `[get]`
- int FileCount `[get]`
- bool enabled `[get, set]`

  Log ON/OFF
- int lastIdx `[get]`

7.21.1 Detailed Description

List / series of single shots

Definition at line 314 of file DataPrimitives.cs.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 ShotList()

```
Axel_hub.ShotList.ShotList (
            bool arch = true,
            string FN = "",
            string prefix = "" )
```

Class constructor if arch -> open file if FN not empty, or create FN if empty if not arch -> ignore FN and prefix

Parameters

arch	log the incomming data
FN	
prefix	

Definition at line 336 of file DataPrimitives.cs.

7.21.3 Member Function Documentation

7.21.3.1 Add()

```
new void Axel_hub.ShotList.Add (
            SingleShot ss )
```

New Add with optional log and size limit

Parameters

ss	

Definition at line 362 of file DataPrimitives.cs.

7.21.3.2 archiScan()

```
SingleShot Axel_hub.ShotList.archiScan (
            out bool next )
```

Get the next scan from a file

Parameters

next	next is false on the last item

Returns

Definition at line 427 of file DataPrimitives.cs.

7.21.3.3 resetScan()

```
void Axel_hub.ShotList.resetScan ( )
```

Reset scan of archive

Definition at line 396 of file DataPrimitives.cs.

7.21.3.4 Save()

```
void Axel_hub.ShotList.Save (
            string FN = "" )
```

Save a file with JSON of single shots per line read the format with ResetScan And ArchiScan

Parameters

FN	

Definition at line 450 of file DataPrimitives.cs.

7.21.4 Member Data Documentation

7.21.4.1 conditions

```
Dictionary<string, double> Axel_hub.ShotList.conditions = new Dictionary<string, double>()
```

Definition at line 322 of file DataPrimitives.cs.

7.21.4.2 depth

```
int Axel_hub.ShotList.depth = 10000  [protected]
```

Definition at line 316 of file DataPrimitives.cs.

7.21.5 Property Documentation

7.21.5.1 archiveMode

```
bool Axel_hub.ShotList.archiveMode  [get]
```

Definition at line 319 of file DataPrimitives.cs.

7.21.5.2 enabled

```
bool Axel_hub.ShotList.enabled  [get], [set]
```

Log ON/OFF

Definition at line 380 of file DataPrimitives.cs.

7.21.5.3 FileCount

```
int Axel_hub.ShotList.FileCount  [get]
```

Definition at line 324 of file DataPrimitives.cs.

7.21.5.4 filename

```
string Axel_hub.ShotList.filename  [get]
```

Definition at line 317 of file DataPrimitives.cs.

7.21.5.5 lastIdx

```
int Axel_hub.ShotList.lastIdx  [get]
```

Definition at line 392 of file DataPrimitives.cs.

7.21.5.6 savingMode

```
bool Axel_hub.ShotList.savingMode  [get]
```

Definition at line 320 of file DataPrimitives.cs.

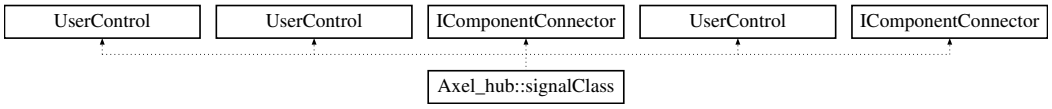
The documentation for this class was generated from the following file:

- DataPrimitives.cs

7.22 Axel_hub::signalClass Class Reference

signalClass

Inheritance diagram for Axel_hub::signalClass:



7.22.1 Detailed Description

signalClass

Interaction logic for signalUC.xaml visualize the raw signal and signal trends {"N1", "N2", "RN1", "RN2", "NTot", "B2", "Btot"} table with last trend position and (optionally) some stats

Definition at line 42 of file signalUC.g.cs.

The documentation for this class was generated from the following files:

- signalUC.g.cs
- signalUC.g.i.cs
- signalUC.xaml.cs

7.23 Axel_hub.SingleShot Class Reference

Class representing single shot with both components quant (MOT) and MEMS (ADC24)

Public Member Functions

- SingleShot ()
  Number of constructors
- SingleShot (double qTime, double qSignal)
- SingleShot (Point q)
- SingleShot (Point q, List< Point > m)
- bool IsEmpty ()
  Check if empty
- int idxByTime (double tm, bool smart=true)
  Find index for specific time
- List< Point > memsPortion (Range< double > rng)
  Get part of mems within a reange
- double memsWeightAccel (double delay, double duration=-1, bool triangle=true)
  Calculating mems acceleration time-related to quant point
- Dictionary< string, double > deconstructAccel (double fringeScale)
  Accelerations components in a dictinary with order, resid, etc.

Public Attributes

- Point quant

Protected Attributes

- string precision = "G5"

Properties

- List< Point > mems `[get, set]`
- string AsString `[get, set]`
  A single shot in JSON format for file import/export

7.23.1 Detailed Description

Class representing single shot with both components quant (MOT) and MEMS (ADC24)

Definition at line 126 of file DataPrimitives.cs.

7.23.2 Constructor & Destructor Documentation

7.23.2.1 SingleShot() [1/4]

```
Axel_hub.SingleShot.SingleShot ( )
```

Number of constructors

Definition at line 137 of file DataPrimitives.cs.

7.23.2.2 SingleShot() [2/4]

```
Axel_hub.SingleShot.SingleShot (
            double qTime,
            double qSignal )
```

Definition at line 142 of file DataPrimitives.cs.

7.23.2.3 SingleShot() [3/4]

```
Axel_hub.SingleShot.SingleShot (
            Point q )
```

Definition at line 147 of file DataPrimitives.cs.

7.23.2.4 SingleShot() [4/4]

```
Axel_hub.SingleShot.SingleShot (
            Point q,
            List< Point > m )
```

Definition at line 152 of file DataPrimitives.cs.

7.23.3 Member Function Documentation

7.23.3.1 deconstructAccel()

```
Dictionary<string, double> Axel_hub.SingleShot.deconstructAccel (
            double fringeScale )
```

Accelerations components in a dictinary with order, resid, etc.

Parameters

fringeScale	

Returns

Definition at line 255 of file DataPrimitives.cs.

7.23.3.2 idxByTime()

```
int Axel_hub.SingleShot.idxByTime (
            double tm,
            bool smart = true )
```

Find index for specific time

Parameters

tm	
smart	More direct way with some assumptions

Returns

Definition at line 174 of file DataPrimitives.cs.

7.23.3.3 IsEmpty()

```
bool Axel_hub.SingleShot.IsEmpty ( )
```

Check if empty

Returns

Definition at line 162 of file DataPrimitives.cs.

7.23.3.4 memsPortion()

```
List<Point> Axel_hub.SingleShot.memsPortion (
            Range< double > rng )
```

Get part of mems within a reange

Parameters

rng	

Returns

Definition at line 201 of file DataPrimitives.cs.

7.23.3.5 memsWeightAccel()

```
double Axel_hub.SingleShot.memsWeightAccel (
            double delay,
            double duration = -1,
            bool triangle = true )
```

Calculating mems acceleration time-related to quant point

Parameters

delay	delay reference to quant.X
duration	the range length
triangle	alternative to triangle is uniform

Returns

Definition at line 216 of file DataPrimitives.cs.

7.23.4 Member Data Documentation

7.23.4.1 precision

```
string Axel_hub.SingleShot.precision = "G5"  [protected]
```

Definition at line 128 of file DataPrimitives.cs.

7.23.4.2 quant

```
Point Axel_hub.SingleShot.quant
```

Definition at line 130 of file DataPrimitives.cs.

7.23.5 Property Documentation

7.23.5.1 AsString

```
string Axel_hub.SingleShot.AsString  [get], [set]
```

A single shot in JSON format for file import/export

Definition at line 282 of file DataPrimitives.cs.

7.23.5.2 mems

```
List<Point> Axel_hub.SingleShot.mems  [get], [set]
```

Definition at line 132 of file DataPrimitives.cs.

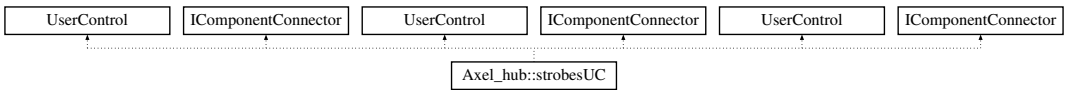
The documentation for this class was generated from the following file:

- DataPrimitives.cs

7.24 Axel_hub::strobesUC Class Reference

strobesUC

Inheritance diagram for Axel_hub::strobesUC:



7.24.1 Detailed Description

strobesUC

Definition at line 42 of file strobeControlUC.g.i.cs.

The documentation for this class was generated from the following files:

- strobeControlUC.g.i.cs
- strobesUC.g.cs
- strobesUC.g.i.cs

Chapter 8

File Documentation

8.1 App.g.cs File Reference

Classes

- class Axel_hub::App

  Interaction logic for App.xaml

Namespaces

- Axel_hub

8.2 App.g.i.cs File Reference

Classes

- class Axel_hub::App

  Interaction logic for App.xaml

Namespaces

- Axel_hub

8.3 App.xaml.cs File Reference

Classes

- class Axel_hub::App

  Interaction logic for App.xaml

Namespaces

- Axel_hub

8.4 AssemblyInfo.cs File Reference

8.5 Axel-hub_Content.g.i.cs File Reference

8.6 AxelAxes.cs File Reference

Classes

- class Axel_hub.AxelAxesClass

  Intermediator between incomming data flow from ucScan user component and AxelAxis user components

Namespaces

- Axel_hub

8.7 AxelAxisUC.g.cs File Reference

Classes

- class Axel_hub::AxelAxisClass

  Interaction logic for AxelAxisUC.xaml AxelAxisClass represents a single axis of acceleration encapsulated and accesable in AxelAxes list of AxelAxisClass Future intermediator abstract movement (linear or rotation) component will be implemented.

Namespaces

- Axel_hub

8.8 AxelAxisUC.g.i.cs File Reference

Classes

- class Axel_hub::AxelAxisClass

  Interaction logic for AxelAxisUC.xaml AxelAxisClass represents a single axis of acceleration encapsulated and accesable in AxelAxes list of AxelAxisClass Future intermediator abstract movement (linear or rotation) component will be implemented.

Namespaces

- Axel_hub

8.9 AxelAxisUC.xaml.cs File Reference

Classes

- class Axel_hub::AxelAxisClass

  Interaction logic for AxelAxisUC.xaml AxelAxisClass represents a single axis of acceleration encapsulated and accesable in AxelAxes list of AxelAxisClass Future intermediator abstract movement (linear or rotation) component will be implemented.

Namespaces

- Axel_hub

8.10 AxelChart.g.i.cs File Reference

Classes

- class Axel_hub::AxelChart

  AxelChart

Namespaces

- Axel_hub

8.11 AxelChartUC.g.cs File Reference

Classes

- class Axel_hub::AxelChartClass

  Interaction logic for AxelChart.xaml

Namespaces

- Axel_hub

8.12 AxelChartUC.g.i.cs File Reference

Classes

- class Axel_hub::AxelChartClass

  Interaction logic for AxelChart.xaml

Namespaces

- Axel_hub

8.13 AxelChartUC.xaml.cs File Reference

Classes

- class Axel_hub::AxelChartClass

  Interaction logic for AxelChart.xaml

Namespaces

- Axel_hub

8.14 AxelHMems.cs File Reference

Classes

- struct Axel_hub.accelCalibr

  Acceleration calibration with optional temperature compensation particular to each MEMS device
- class Axel_hub.AxelMems

  The hardware abstraction for MEMS with ADC24 (NI9251) device
- class Axel_hub.AxelMemsTemperature

  The temperature in a class abstraction

Namespaces

- Axel_hub

8.15 DataPrimitives.cs File Reference

Classes

- class **Axel_hub.MMDataConverter**

  Averaging the photo diode signals {"N2", "NTot", "B2", "BTot", "Bg"}
- class Axel_hub.SingleShot

  Class representing single shot with both components quant (MOT) and MEMS (ADC24)
- class Axel_hub.ShotList

  List / series of single shots

Namespaces

- Axel_hub

8.16 DataStackLib.cs File Reference

Classes

- class Axel_hub.DataStack

  You (developer) need to set TimeMode and one of SizeLimit or TimeLimit TimeMode is about the way DataStack limits its size The output is from standart List method ToArray in order to set DataSource of Graph

Namespaces

- Axel_hub

8.17 GeneratedInternalTypeHelper.g.cs File Reference

8.18 GeneratedInternalTypeHelper.g.i.cs File Reference

Classes

- class XamlGeneratedNamespace::GeneratedInternalTypeHelper

  GeneratedInternalTypeHelper

Namespaces

- XamlGeneratedNamespace

8.19 JoinOptimUC.g.i.cs File Reference

Classes

- class Axel_hub::JoinOptimClass

  JoinOptimClass

Namespaces

- Axel_hub

8.20 MainWindow.g.cs File Reference

Classes

- class Axel_hub::MainWindow

  Interaction logic for MainWindow.xaml command line arguments (space separated): -remote:`partner` *-hw↩ :*`config.file` *where* `partner` *is remote partner name* `title`*; hw<c>hardware,* `config.file.hw` *is in Config folder*

Namespaces

- Axel_hub

8.21 MainWindow.g.i.cs File Reference

Classes

- class Axel_hub::MainWindow

  Interaction logic for MainWindow.xaml command line arguments (space separated): -remote:`partner` *-hw:*`config.file` *where* `partner` *is remote partner name* `title`*; hw<c>hardware,* `config.file.hw` *is in Config folder*

Namespaces

- Axel_hub

8.22 MainWindow.xaml.cs File Reference

Classes

- class Axel_hub::MainWindow

  Interaction logic for MainWindow.xaml command line arguments (space separated): -remote:`partner` *-hw:*`config.file` *where* `partner` *is remote partner name* `title`*; hw<c>hardware,* `config.file.hw` *is in Config folder*

Namespaces

- Axel_hub

Functions

- public delegate void Axel_hub::StartDelegate ()

8.23 Options.g.cs File Reference

Classes

- class OptionsNS::OptionsWindow

  OptionsWindow

Namespaces

- OptionsNS

8.24 Options.g.i.cs File Reference

Classes

- class OptionsNS::OptionsWindow

  OptionsWindow

Namespaces

- OptionsNS

8.25 Options.g.i.cs File Reference

Classes

- class OptionsNS::OptionsWindow

  OptionsWindow

Namespaces

- OptionsNS

8.26 Options.xaml.cs File Reference

Classes

- class OptionsNS::OptionsWindow

  OptionsWindow

Namespaces

- OptionsNS

8.27 OptionsType.cs File Reference

Classes

- class OptionsNS.GeneralOptions

  general options from Options dialog window accesable everywhere
- class OptionsNS.ScanModes

  visuals for the app, MEMS aqcuisition params and scan modes
- class OptionsNS.Modes

  Visuals and prameters for Top: Axel-chart Middle: Signal panel charts Bottom: Scan and Accel trend tabs/charts

Namespaces

- OptionsNS

Enumerations

- enum OptionsNS.RemoteMode {
  OptionsNS.RemoteMode.Disconnected, OptionsNS.RemoteMode.Jumbo_Scan, OptionsNS.RemoteMode.Jumbo_Repeat,
  OptionsNS.RemoteMode.Simple_Scan,
  OptionsNS.RemoteMode.Simple_Repeat, OptionsNS.RemoteMode.Ready_To_Remote }

  The mode negotiated with MM2

8.28 README.md File Reference

8.29 Resources.Designer.cs File Reference

Classes

- class Axel_hub::Properties::Resources

  A strongly-typed resource class, for looking up localized strings, etc.

Namespaces

- Axel_hub
- Axel_hub::Properties

8.30 scanUC.g.cs File Reference

Classes

- class Axel_hub::scanClass

  scanClass

Namespaces

- Axel_hub

8.31 scanUC.g.i.cs File Reference

Classes

- class Axel_hub::scanClass

  scanClass

Namespaces

- Axel_hub

8.32 scanUC.xaml.cs File Reference

Classes

- struct Axel_hub::FringeParams

  fringes(phi) = cos(period ∗ phi + phase) + offset

Namespaces

- Axel_hub

Functions

- public Axel_hub::scanClass ()

  Class constructor - set defaults
- public void Axel_hub::InitOptions (ref GeneralOptions _genOptions, ref ScanModes _scanModes)

  Initialize - set genOptions
- public void Axel_hub::UpdateModes ()

  Set internal from visual modes
- public bool Axel_hub::SendJson (string json, bool async=false)

  Wrapper of remote.sendCommand
- public void Axel_hub::SetActivity (string act)
- public void Axel_hub::SetSamplingRate (int rate)
- public void Axel_hub::SetFringeParams (FringeParams fp)

  Show fringes params
- public void Axel_hub::OnRealSampling (double _realSampling)
- private void Axel_hub::dispatcherTimer_Tick (object sender, EventArgs e)

  Shows visual progress of ADC24 acquisition
- private void Axel_hub::Status (string sts)
- protected void Axel_hub::RemoteModeEvent (RemoteMode oldMode, RemoteMode newMode)
- private bool Axel_hub::OnReceive (string message)

  Incomming from MM2/Axel-probe message
- public delegate void Axel_hub::StartHandler (bool jumbo, bool down, double period, int sizeLimit)

  Start/Stop group operation wity ADC24 params
- protected void Axel_hub::StartEvent (bool jumbo, bool down, double period, int sizeLimit)
- public delegate void Axel_hub::RemoteHandler (string msg)

  Incomming message event thingy
- protected void Axel_hub::RemoteEvent (string msg)
- public delegate void Axel_hub::LogHandler (string txt, Color? clr=null)

  Log into left text box
- protected void Axel_hub::LogEvent (string txt, Color? clr=null)
- protected void Axel_hub::OnAsyncSend (bool OK, string json2send)

  Report sent message in log
- private void Axel_hub::Image_MouseDown (object sender, MouseButtonEventArgs e)
- public double Axel_hub::GetSamplingPeriod ()

Get the sampling period regardless the units

- public int Axel_hub::GetBufferSize ()

  Get the buffer size depending of settings

- protected void Axel_hub::ActiveRemote (bool active)
- private void Axel_hub::OnActiveComm (bool active, bool forced)

  Event when the connection goes ON/OFF

- private void Axel_hub::UserControl_Loaded (object sender, RoutedEventArgs e)

  Some secondary to contructor initialilzations

Variables

- public struct Axel_hub::FringeParams Axel_hub::realSampling

  Interaction logic for UserControl1.xaml

- private string Axel_hub::ArrangedPartner = ""
- TimeSpan Axel_hub::totalTime
- TimeSpan Axel_hub::currentTime
- public DispatcherTimer Axel_hub::dTimer
- GeneralOptions Axel_hub::genOptions = null
- public ScanModes Axel_hub::scanModes = null
- private bool Axel_hub::_Running
- public bool Axel_hub::Running

  Some visual adjustments when ADC24 starts/stops

- public RemoteMode Axel_hub::remoteMode

  Current remode mode - defines the context next group shots

- public event RemoteModeHandler Axel_hub::OnRemoteMode
- public RemoteMessaging Axel_hub::remote { get
- Axel_hub::set
- public event StartHandler Axel_hub::OnStart
- public event RemoteHandler Axel_hub::OnRemote
- public event LogHandler Axel_hub::OnLog
- private bool Axel_hub::_jumboButton = true
- private bool Axel_hub::jumboButton

  Set the main scan button to Jumbo mode

- public event ActiveRemoteHandler Axel_hub::OnActiveRemote

8.33 Settings.Designer.cs File Reference

Classes

- class Axel_hub::Properties::Settings

Namespaces

- Axel_hub
- Axel_hub::Properties

8.34 signalUC.g.cs File Reference

Classes

- class Axel_hub::signalClass
  signalClass

Namespaces

- Axel_hub

8.35 signalUC.g.i.cs File Reference

Classes

- class Axel_hub::signalClass
  signalClass

Namespaces

- Axel_hub

8.36 signalUC.xaml.cs File Reference

Classes

- class Axel_hub::signalClass
  signalClass

Namespaces

- Axel_hub

8.37 strobeControlUC.g.i.cs File Reference

Classes

- class Axel_hub::strobesUC
  strobesUC

Namespaces

- Axel_hub

8.38 strobesUC.g.cs File Reference

Classes

- class Axel_hub::strobesUC

  strobesUC

Namespaces

- Axel_hub

8.39 strobesUC.g.i.cs File Reference

Classes

- class Axel_hub::strobesUC

  strobesUC

Namespaces

- Axel_hub

8.40 strobesUC.xaml.cs File Reference

Classes

- class **Axel_hub::calcAccel**

  Library for calculating acceleration from fringes, phase, etc

Namespaces

- Axel_hub

Functions

- public Axel_hub::strobesUC ()

  Class constructor
- public void Axel_hub::Reset ()

  Initaile strobe for axel-probe simulated fringes
- public void Axel_hub::Flip ()

  Exchange UP/DOWN strobe positions
- public void Axel_hub::Init (string _prefix)

  Initiate strobe from file settings
- public void Axel_hub::OnJumboRepeat (double _fringeScale, double _fringeShift, MMexec _grpMME, double contrastV)

  Call this before each Jumbo Repeat for group MMexec and modes synchronization
- public delegate void Axel_hub::LogHandler (string txt, Color? clr=null)

  Log into left text box
- protected void Axel_hub::LogEvent (string txt, Color? clr=null)
- public double Axel_hub::centreFringe ()

  Calculating fringe centre
- public double Axel_hub::calcContrast (double A)

  Calculating contrast
- public double Axel_hub::zeroFringe ()

  Calculating zeroFringe - similar to centreFring but woth phase shift compensation
- public Dictionary< string, double > Axel_hub::deconstructAccel (double accel, double mems)

  Deconstructing accaleration to acceleration components - see dictionary keys
- public double Axel_hub::nextShot (int runID, double asymmetry, out double correction)

  Calculated phaseCorr - corrected Raman phase (0 if not PID)
- public MMexec Axel_hub::backMME (int runID, double asymmetry, MMexec mme=null)

  Prepare back message with new Raman phase value
- private void Axel_hub::fillReport (Dictionary< string, double > rpr)

  Update table with strobes/PID calculation results
- public double Axel_hub::PID (double disbalance)

  Calculating the phase correction from the disbalance on strobes Ys
- public void Axel_hub::SaveConfigFile ()

  Save Config file in Config directory of Axel-hub
- public void Axel_hub::OpenConfigFile ()

  Open Config file from Config directory of Axel-hub

Variables

- private MMexec Axel_hub::grpMME
- private MMexec Axel_hub::lastMMEin
- private MMexec Axel_hub::lastMMEout
- private bool Axel_hub::_PID_Enabled
- public bool Axel_hub::PID_Enabled

  PID follow the strobe position
- private double Axel_hub::lastContrast = -1
- int Axel_hub::runI = 0
- string Axel_hub::configFile
- List< double > Axel_hub::iStack
- List< double > Axel_hub::dStack
- private FileLogger Axel_hub::logger
- public Point Axel_hub::Down
- public Point Axel_hub::Up
- public Point Axel_hub::Low
- string[ ] Axel_hub::Titles = { "runI", "tP", "tI", "tD", "Down.X", "Up.X", "disbal", "corr", "iSD-R", "contrast" }

Index

_PID_Enabled
    Axel_hub, 28
_Running
    Axel_hub, 29
_jumboButton
    Axel_hub, 28

accel
    Axel_hub.accelCalibr, 37
AcquireEvent
    Axel_hub.AxelMems, 52
AcquireHandler
    Axel_hub.AxelMems, 52
activeChannel
    Axel_hub.AxelMems, 57
ActiveRemote
    Axel_hub, 16
Add
    Axel_hub.DataStack, 62
    Axel_hub.ShotList, 97
AddAxis
    Axel_hub.AxelAxesClass, 42
AddPoint
    Axel_hub.DataStack, 62
AddRange
    Axel_hub.DataStack, 63
AdjustTimelineToStopwatch
    Axel_hub.AxelMems, 55
App.g.cs, 107
App.g.i.cs, 107
App.xaml.cs, 107
archiScan
    Axel_hub.ShotList, 97
archiveMode
    Axel_hub.ShotList, 99
ArrangedPartner
    Axel_hub, 29
ask
    OptionsNS.GeneralOptions, 77
AssemblyInfo.cs, 108
AsString
    Axel_hub.SingleShot, 104
AutoScaleBottom
    OptionsNS.Modes, 85
AutoScaleMiddle
    OptionsNS.Modes, 85
Axel-hub_Content.g.i.cs, 108
Axel_hub, 13
    _PID_Enabled, 28
    _Running, 29
    _jumboButton, 28
    ActiveRemote, 16
    ArrangedPartner, 29
    backMME, 16
    calcContrast, 17
    centreFringe, 17
    configFile, 29
    currentTime, 29
    deconstructAccel, 17
    dispatcherTimer_Tick, 19
    Down, 29
    dStack, 29
    dTimer, 30
    fillReport, 19
    Flip, 19
    genOptions, 30
    GetBufferSize, 20
    GetSamplingPeriod, 20
    grpMME, 30
    Image_MouseDown, 20
    Init, 20
    InitOptions, 21
    iStack, 30
    jumboButton, 30
    lastContrast, 30
    lastMMEin, 31
    lastMMEout, 31
    LogEvent, 21
    logger, 31
    LogHandler, 21
    Low, 31
    nextShot, 22
    OnActiveComm, 22
    OnActiveRemote, 31
    OnAsyncSend, 22
    OnJumboRepeat, 23
    OnLog, 31
    OnRealSampling, 23
    OnReceive, 23
    OnRemote, 32
    OnRemoteMode, 32
    OnStart, 32
    OpenConfigFile, 24
    PID, 24
    PID_Enabled, 32
    realSampling, 32
    remote, 33
    RemoteEvent, 24
    RemoteHandler, 24

remoteMode, 33
RemoteModeEvent, 25
Reset, 25
runI, 33
Running, 33
SaveConfigFile, 25
scanClass, 25
scanModes, 33
SendJson, 25
set, 34
SetActivity, 26
SetFringeParams, 26
SetSamplingRate, 26
StartDelegate, 26
StartEvent, 27
StartHandler, 27
Status, 27
strobesUC, 27
Titles, 34
totalTime, 34
Up, 34
UpdateModes, 27
UserControl_Loaded, 28
zeroFringe, 28
Axel_hub.accelCalibr, 37
accel, 37
cK0, 38
cK1, 38
model, 38
pK0, 38
pK1, 38
rAccel, 39
rTemper, 39
SN, 39
Axel_hub.AxelAxesClass, 40
AddAxis, 42
AxelAxesClass, 41
axelMems, 47
byName, 42
Clear, 42
Closing, 43
DoAcquire, 43
DoAcquireTemperature, 44
DoJumboScan, 44
DoRemote, 44
jumboRepeat, 44
LogEvent, 45
LogHandler, 45
memsRunning, 47
OnLog, 48
prfIdx, 45
rCount, 48
SaveDefaultModes, 46
set2startADC24, 46
SetChartStrobes, 46
startADC, 46
UpdateFromOptions, 47
Axel_hub.AxelMems, 50
AcquireEvent, 52
AcquireHandler, 52
activeChannel, 57
AdjustTimelineToStopwatch, 55
AxelMems, 52
byADCtimer, 51
byBoth, 51
byNone, 51
byStopwatch, 51
configureVITask, 52
FixConvRate, 55
hw, 56
isDevicePlugged, 53
memsX, 56
nSamples, 57
OnAcquire, 57
OnRealSampling, 57
rawData, 56
readBurst, 53
RealConvRate, 53
RealSamplingEvent, 54
RealSamplingHandler, 54
Reset, 54
running, 57
sampleRate, 57
SetStopwatch, 54
StartAcquisition, 54
StartStopwatch, 55
StopAcquisition, 55
TimeElapsed, 55
Timeout, 56
TimingMode, 56
TimingModes, 51
Axel_hub.AxelMemsTemperature, 58
AxelMemsTemperature, 58
hw, 59
TakeTheTemperature, 58
Axel_hub.DataStack, 59
Add, 62
AddPoint, 62
AddRange, 63
Clear, 63
Clone, 63
Compress, 64
CopyEach, 64
DataStack, 61
Depth, 73
ExportToArray, 64
fillSamples, 65
First, 73
Fit2Limit, 65
generalIdx, 72
ImportFromArray, 65
importFromArrays, 66
indexByX, 66
Last, 73
lastError, 73
logger, 72

maxDepth, 72
OnRefresh, 74
OpenPair, 66
pointSDev, 67
pointXs, 67
pointYs, 67
Portion, 68
prefix, 73
RefFileStats, 72
RefreshEvent, 68
RefreshHandler, 68
rem, 74
Rescale, 68
Running, 74
SavePair, 70
StackMode, 74
statsByIdx, 70
statsByTime, 71
stopWatch, 72
TimePortion, 71
TimeSeriesMode, 74
visualCountLimit, 72
Axel_hub.ShotList, 96
Add, 97
archiScan, 97
archiveMode, 99
conditions, 98
depth, 98
enabled, 99
FileCount, 99
filename, 99
lastIdx, 99
resetScan, 98
Save, 98
savingMode, 100
ShotList, 97
Axel_hub.SingleShot, 100
AsString, 104
deconstructAccel, 102
idxByTime, 102
IsEmpty, 103
mems, 105
memsPortion, 103
memsWeightAccel, 104
precision, 104
quant, 104
SingleShot, 101, 102
Axel_hub::App, 39
Axel_hub::AxelAxisClass, 48
Axel_hub::AxelChart, 49
Axel_hub::AxelChartClass, 49
Axel_hub::FringeParams, 75
offset, 75
period, 75
phase, 76
Axel_hub::JoinOptimClass, 82
Axel_hub::MainWindow, 83
Axel_hub::Properties, 34
Axel_hub::Properties::Resources, 91
Axel_hub::Properties::Settings, 95
Axel_hub::scanClass, 92
Axel_hub::signalClass, 100
Axel_hub::strobesUC, 105
AxelAxes.cs, 108
AxelAxesClass
Axel_hub.AxelAxesClass, 41
AxelAxisUC.g.cs, 108
AxelAxisUC.g.i.cs, 108
AxelAxisUC.xaml.cs, 109
AxelChart.g.i.cs, 109
AxelChartUC.g.cs, 109
AxelChartUC.g.i.cs, 109
AxelChartUC.xaml.cs, 110
AxelHMems.cs, 110
AxelMems
Axel_hub.AxelMems, 52
axelMems
Axel_hub.AxelAxesClass, 47
AxelMemsTemperature
Axel_hub.AxelMemsTemperature, 58
AxesChannels
OptionsNS.GeneralOptions, 78

Background
OptionsNS.Modes, 85
backMME
Axel_hub, 16
byADCtimer
Axel_hub.AxelMems, 51
byBoth
Axel_hub.AxelMems, 51
byName
Axel_hub.AxelAxesClass, 42
byNone
Axel_hub.AxelMems, 51
byStopwatch
Axel_hub.AxelMems, 51

calcContrast
Axel_hub, 17
centreFringe
Axel_hub, 17
ChartUpdate
OptionsNS.Modes, 85
cK0
Axel_hub.accelCalibr, 38
cK1
Axel_hub.accelCalibr, 38
Clear
Axel_hub.AxelAxesClass, 42
Axel_hub.DataStack, 63
Clone
Axel_hub.DataStack, 63
Closing
Axel_hub.AxelAxesClass, 43
Compress
Axel_hub.DataStack, 64

conditions
Axel_hub.ShotList, 98
configFile
Axel_hub, 29
configureVITask
Axel_hub.AxelMems, 52
CopyEach
Axel_hub.DataStack, 64
currentTime
Axel_hub, 29

DarkCurrent
OptionsNS.Modes, 85
DataPrimitives.cs, 110
DataStack
Axel_hub.DataStack, 61
DataStackLib.cs, 111
deconstructAccel
Axel_hub, 17
Axel_hub.SingleShot, 102
Depth
Axel_hub.DataStack, 73
depth
Axel_hub.ShotList, 98
Disconnected
OptionsNS, 35
dispatcherTimer_Tick
Axel_hub, 19
DoAcquire
Axel_hub.AxelAxesClass, 43
DoAcquireTemperature
Axel_hub.AxelAxesClass, 44
DoJumboScan
Axel_hub.AxelAxesClass, 44
DoRemote
Axel_hub.AxelAxesClass, 44
DoubleStrobe
OptionsNS.Modes, 85
Down
Axel_hub, 29
dStack
Axel_hub, 29
dTimer
Axel_hub, 30

enabled
Axel_hub.ShotList, 99
ExportToArray
Axel_hub.DataStack, 64

FileCount
Axel_hub.ShotList, 99
filename
Axel_hub.ShotList, 99
fillReport
Axel_hub, 19
fillSamples
Axel_hub.DataStack, 65
First
Axel_hub.DataStack, 73
Fit2Limit
Axel_hub.DataStack, 65
FixConvRate
Axel_hub.AxelMems, 55
Flip
Axel_hub, 19
followPID
OptionsNS.GeneralOptions, 78

generalIdx
Axel_hub.DataStack, 72
GeneratedInternalTypeHelper.g.cs, 111
GeneratedInternalTypeHelper.g.i.cs, 111
genOptions
Axel_hub, 30
GetBufferSize
Axel_hub, 20
GetSamplingPeriod
Axel_hub, 20
grpMME
Axel_hub, 30

Height
OptionsNS.ScanModes, 93
hw
Axel_hub.AxelMems, 56
Axel_hub.AxelMemsTemperature, 59

idxByTime
Axel_hub.SingleShot, 102
Image_MouseDown
Axel_hub, 20
ImportFromArray
Axel_hub.DataStack, 65
importFromArrays
Axel_hub.DataStack, 66
indexByX
Axel_hub.DataStack, 66
Init
Axel_hub, 20
InitOptions
Axel_hub, 21
intN2
OptionsNS.GeneralOptions, 78
isDevicePlugged
Axel_hub.AxelMems, 53
IsEmpty
Axel_hub.SingleShot, 103
iStack
Axel_hub, 30

JoinLog
OptionsNS.Modes, 86
JoinOptimUC.g.i.cs, 111
Jumbo_Repeat
OptionsNS, 35
Jumbo_Scan
OptionsNS, 35

jumboButton
Axel_hub, 30
JumboBy
OptionsNS.Modes, 86
JumboCycles
OptionsNS.Modes, 86
JumboFrom
OptionsNS.Modes, 86
JumboRepeat
OptionsNS.GeneralOptions, 78
jumboRepeat
Axel_hub.AxelAxesClass, 44
JumboScan
OptionsNS.GeneralOptions, 79
JumboTo
OptionsNS.Modes, 86

Kcoeff
OptionsNS.Modes, 86
kD
OptionsNS.Modes, 87
kI
OptionsNS.Modes, 87
kP
OptionsNS.Modes, 87

Last
Axel_hub.DataStack, 73
lastContrast
Axel_hub, 30
lastError
Axel_hub.DataStack, 73
lastIdx
Axel_hub.ShotList, 99
lastMMEin
Axel_hub, 31
lastMMEout
Axel_hub, 31
Left
OptionsNS.ScanModes, 94
LogEvent
Axel_hub, 21
Axel_hub.AxelAxesClass, 45
LogFilePrec
OptionsNS.GeneralOptions, 79
logger
Axel_hub, 31
Axel_hub.DataStack, 72
LogHandler
Axel_hub, 21
Axel_hub.AxelAxesClass, 45
Low
Axel_hub, 31

MainWindow.g.cs, 111
MainWindow.g.i.cs, 112
MainWindow.xaml.cs, 112
maxDepth
Axel_hub.DataStack, 72
mems
Axel_hub.SingleShot, 105
Mems2SignDelay
OptionsNS.GeneralOptions, 79
Mems2SignLen
OptionsNS.GeneralOptions, 79
MemsEnabled
OptionsNS.Modes, 87
MemsHw
OptionsNS.GeneralOptions, 79
MemsInJumbo
OptionsNS.GeneralOptions, 79
memsPortion
Axel_hub.SingleShot, 103
memsRunning
Axel_hub.AxelAxesClass, 47
memsWeightAccel
Axel_hub.SingleShot, 104
memsX
Axel_hub.AxelMems, 56
MiddleFrame
OptionsNS.Modes, 87
model
Axel_hub.accelCalibr, 38

N1
OptionsNS.Modes, 87
N2
OptionsNS.Modes, 88
nextShot
Axel_hub, 22
nosave
OptionsNS.GeneralOptions, 77
nSamples
Axel_hub.AxelMems, 57
Ntot
OptionsNS.Modes, 88

offset
Axel_hub::FringeParams, 75
OptionsNS.Modes, 88
OnAcquire
Axel_hub.AxelMems, 57
OnActiveComm
Axel_hub, 22
OnActiveRemote
Axel_hub, 31
OnAsyncSend
Axel_hub, 22
OnJumboRepeat
Axel_hub, 23
OnLog
Axel_hub, 31
Axel_hub.AxelAxesClass, 48
OnRealSampling
Axel_hub, 23
Axel_hub.AxelMems, 57
OnReceive
Axel_hub, 23

OnRefresh
Axel_hub.DataStack, 74
OnRemote
Axel_hub, 32
OnRemoteMode
Axel_hub, 32
OnStart
Axel_hub, 32
OpenConfigFile
Axel_hub, 24
OpenPair
Axel_hub.DataStack, 66
Options.g.cs, 112
Options.g.i.cs, 113
Options.xaml.cs, 113
OptionsNS, 35
Disconnected, 35
Jumbo_Repeat, 35
Jumbo_Scan, 35
Ready_To_Remote, 35
RemoteMode, 35
Simple_Repeat, 35
Simple_Scan, 35
OptionsNS.GeneralOptions, 76
ask, 77
AxesChannels, 78
followPID, 78
intN2, 78
JumboRepeat, 78
JumboScan, 79
LogFilePrec, 79
Mems2SignDelay, 79
Mems2SignLen, 79
MemsHw, 79
MemsInJumbo, 79
nosave, 77
RawSignalAvg, 80
Save, 77
save, 77
SaveFilePrec, 80
SaveModes, 77
saveModes, 78
saveVisuals, 80
ShowMemsIfRunning, 80
SignalCursorPrec, 80
SignalTablePrec, 80
TemperatureCompensation, 81
TemperatureEnabled, 81
TemperatureHw, 81
TrendSignalLen, 81
OptionsNS.Modes, 83
AutoScaleBottom, 85
AutoScaleMiddle, 85
Background, 85
ChartUpdate, 85
DarkCurrent, 85
DoubleStrobe, 85
JoinLog, 86
JumboBy, 86
JumboCycles, 86
JumboFrom, 86
JumboTo, 86
Kcoeff, 86
kD, 87
kI, 87
kP, 87
MemsEnabled, 87
MiddleFrame, 87
N1, 87
N2, 88
Ntot, 88
offset, 88
phi0, 88
PID_Enabled, 88
PowerCoeff, 88
RN1, 89
RN2, 89
RollMean, 89
RsltChrtUpdate, 89
RsltTblUpdate, 89
Save, 84
scale, 89
ShowFreq, 90
SignalLog, 90
StackDepth, 90
StdDev, 90
TblUpdate, 90
TopFrame, 90
TopOfTopFrame, 91
OptionsNS.ScanModes, 92
Height, 93
Left, 94
remoteMode, 93
SamplingFreq, 94
Save, 93
SizeLimit, 94
SizeLimitMode, 94
TimeLimit, 94
TimeLimitMode, 94
Top, 95
Width, 95
OptionsNS::OptionsWindow, 91
OptionsType.cs, 113

period
Axel_hub::FringeParams, 75
phase
Axel_hub::FringeParams, 76
phi0
OptionsNS.Modes, 88
PID
Axel_hub, 24
PID_Enabled
Axel_hub, 32
OptionsNS.Modes, 88
pK0
Axel_hub.accelCalibr, 38

pK1
Axel_hub.accelCalibr, 38
pointSDev
Axel_hub.DataStack, 67
pointXs
Axel_hub.DataStack, 67
pointYs
Axel_hub.DataStack, 67
Portion
Axel_hub.DataStack, 68
PowerCoeff
OptionsNS.Modes, 88
precision
Axel_hub.SingleShot, 104
prefix
Axel_hub.DataStack, 73
prfIdx
Axel_hub.AxelAxesClass, 45

quant
Axel_hub.SingleShot, 104

rAccel
Axel_hub.accelCalibr, 39
rawData
Axel_hub.AxelMems, 56
RawSignalAvg
OptionsNS.GeneralOptions, 80
rCount
Axel_hub.AxelAxesClass, 48
readBurst
Axel_hub.AxelMems, 53
README.md, 114
Ready_To_Remote
OptionsNS, 35
RealConvRate
Axel_hub.AxelMems, 53
realSampling
Axel_hub, 32
RealSamplingEvent
Axel_hub.AxelMems, 54
RealSamplingHandler
Axel_hub.AxelMems, 54
RefFileStats
Axel_hub.DataStack, 72
RefreshEvent
Axel_hub.DataStack, 68
RefreshHandler
Axel_hub.DataStack, 68
rem
Axel_hub.DataStack, 74
remote
Axel_hub, 33
RemoteEvent
Axel_hub, 24
RemoteHandler
Axel_hub, 24
RemoteMode
OptionsNS, 35
remoteMode
Axel_hub, 33
OptionsNS.ScanModes, 93
RemoteModeEvent
Axel_hub, 25
Rescale
Axel_hub.DataStack, 68
Reset
Axel_hub, 25
Axel_hub.AxelMems, 54
resetScan
Axel_hub.ShotList, 98
Resources.Designer.cs, 114
RN1
OptionsNS.Modes, 89
RN2
OptionsNS.Modes, 89
RollMean
OptionsNS.Modes, 89
RsltChrtUpdate
OptionsNS.Modes, 89
RsltTblUpdate
OptionsNS.Modes, 89
rTemper
Axel_hub.accelCalibr, 39
runI
Axel_hub, 33
Running
Axel_hub, 33
Axel_hub.DataStack, 74
running
Axel_hub.AxelMems, 57

sampleRate
Axel_hub.AxelMems, 57
SamplingFreq
OptionsNS.ScanModes, 94
Save
Axel_hub.ShotList, 98
OptionsNS.GeneralOptions, 77
OptionsNS.Modes, 84
OptionsNS.ScanModes, 93
save
OptionsNS.GeneralOptions, 77
SaveConfigFile
Axel_hub, 25
SaveDefaultModes
Axel_hub.AxelAxesClass, 46
SaveFilePrec
OptionsNS.GeneralOptions, 80
SaveModes
OptionsNS.GeneralOptions, 77
saveModes
OptionsNS.GeneralOptions, 78
SavePair
Axel_hub.DataStack, 70
saveVisuals
OptionsNS.GeneralOptions, 80
savingMode

Axel_hub.ShotList, 100
scale
Options NS.Modes, 89
scanClass
Axel_hub, 25
scanModes
Axel_hub, 33
scanUC.g.cs, 114
scanUC.g.i.cs, 114
scanUC.xaml.cs, 115
SendJson
Axel_hub, 25
set
Axel_hub, 34
set2startADC24
Axel_hub.AxelAxesClass, 46
SetActivity
Axel_hub, 26
SetChartStrobes
Axel_hub.AxelAxesClass, 46
SetFringeParams
Axel_hub, 26
SetSamplingRate
Axel_hub, 26
SetStopwatch
Axel_hub.AxelMems, 54
Settings.Designer.cs, 116
ShotList
Axel_hub.ShotList, 97
ShowFreq
Options NS.Modes, 90
ShowMemsIfRunning
Options NS.GeneralOptions, 80
SignalCursorPrec
Options NS.GeneralOptions, 80
SignalLog
Options NS.Modes, 90
SignalTablePrec
Options NS.GeneralOptions, 80
signalUC.g.cs, 117
signalUC.g.i.cs, 117
signalUC.xaml.cs, 117
Simple_Repeat
Options NS, 35
Simple_Scan
Options NS, 35
SingleShot
Axel_hub.SingleShot, 101, 102
SizeLimit
Options NS.ScanModes, 94
SizeLimitMode
Options NS.ScanModes, 94
SN
Axel_hub.accelCalibr, 39
StackDepth
Options NS.Modes, 90
StackMode
Axel_hub.DataStack, 74
StartAcquisition
Axel_hub.AxelMems, 54
startADC
Axel_hub.AxelAxesClass, 46
StartDelegate
Axel_hub, 26
StartEvent
Axel_hub, 27
StartHandler
Axel_hub, 27
StartStopwatch
Axel_hub.AxelMems, 55
statsByIdx
Axel_hub.DataStack, 70
statsByTime
Axel_hub.DataStack, 71
Status
Axel_hub, 27
StdDev
Options NS.Modes, 90
StopAcquisition
Axel_hub.AxelMems, 55
stopWatch
Axel_hub.DataStack, 72
strobeControlUC.g.i.cs, 117
strobesUC
Axel_hub, 27
strobesUC.g.cs, 118
strobesUC.g.i.cs, 118
strobesUC.xaml.cs, 118

TakeTheTemperature
Axel_hub.AxelMemsTemperature, 58
TblUpdate
Options NS.Modes, 90
TemperatureCompensation
Options NS.GeneralOptions, 81
TemperatureEnabled
Options NS.GeneralOptions, 81
TemperatureHw
Options NS.GeneralOptions, 81
TimeElapsed
Axel_hub.AxelMems, 55
TimeLimit
Options NS.ScanModes, 94
TimeLimitMode
Options NS.ScanModes, 94
Timeout
Axel_hub.AxelMems, 56
TimePortion
Axel_hub.DataStack, 71
TimeSeriesMode
Axel_hub.DataStack, 74
TimingMode
Axel_hub.AxelMems, 56
TimingModes
Axel_hub.AxelMems, 51
Titles
Axel_hub, 34

Top
    OptionsNS.ScanModes, 95
TopFrame
    OptionsNS.Modes, 90
TopOfTopFrame
    OptionsNS.Modes, 91
totalTime
    Axel_hub, 34
TrendSignalLen
    OptionsNS.GeneralOptions, 81

Up
    Axel_hub, 34
UpdateFromOptions
    Axel_hub.AxelAxesClass, 47
UpdateModes
    Axel_hub, 27
UserControl_Loaded
    Axel_hub, 28

visualCountLimit
    Axel_hub.DataStack, 72

Width
    OptionsNS.ScanModes, 95

XamlGeneratedNamespace, 35
XamlGeneratedNamespace::GeneratedInternalTypeHelper,
        82

zeroFringe
    Axel_hub, 28