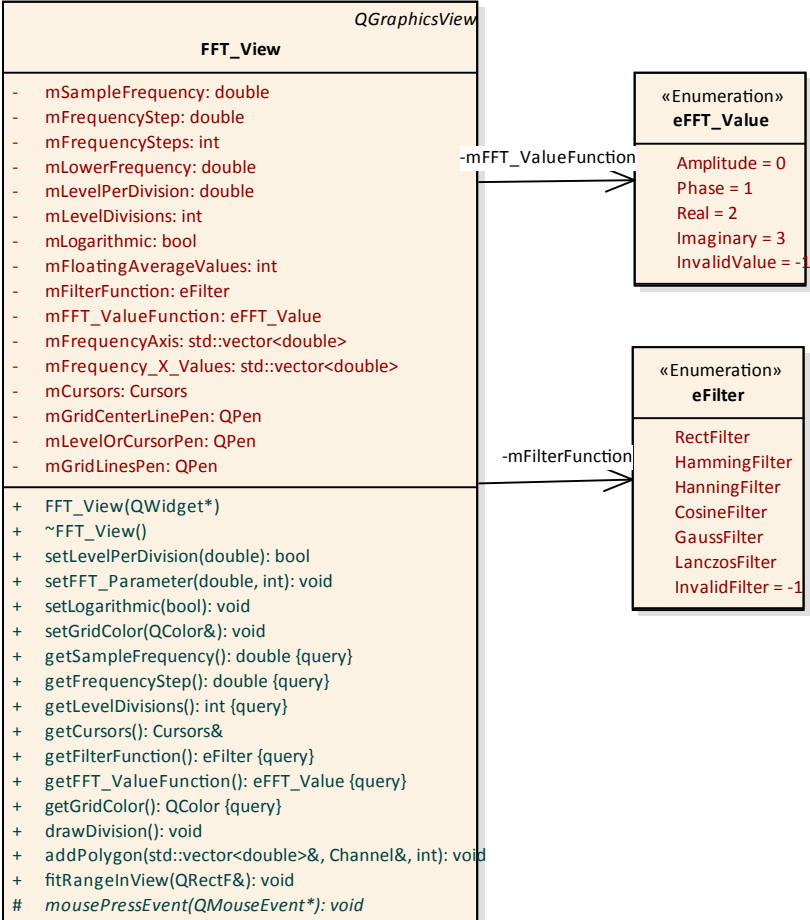


ScopeView	QGraphicsView
<div><div>- mStartTime: double</div><div>- mTimeStep: double</div><div>- mTimePerDivision: double</div><div>- mLevelPerDivision: double</div><div>- mTriggerLevel: double</div><div>- mTimeDivisions: int</div><div>- mLevelDivisions: int</div><div>- mLissajousView: bool</div><div>- mCursors: Cursors</div><div>- mGridCenterLinePen: QPen</div><div>- mLevelOrCursorPen: QPen</div><div>- mGridLinesPen: QPen</div></div>	
<div><div>+ ScopeView(QWidget*)</div><div>+ ~ScopeView()</div><div>+ setLevelPerDivision(double): bool</div><div>+ setTimePerDivision(double): void</div><div>+ setLissajousView(bool): void</div><div>+ setTimeStep(double): void</div><div>+ setTriggerStartTime(double): void</div><div>+ setTriggerLevel(double): void</div><div>+ setGridColor(QColor&amp;): void</div><div>+ getTotalTime(): double {query}</div><div>+ getTimeStep(): double {query}</div><div>+ getTriggerStartTime(): double {query}</div><div>+ getTriggerLevel(): double {query}</div><div>+ getLevelDivisions(): int {query}</div><div>+ getTimeDivisions(): int {query}</div><div>+ getDrawnTimeSteps(): int {query}</div><div>+ getGridColor(): QColor {query}</div><div>+ getCursors(): Cursors&amp;</div><div>+ isLissayousView(): bool</div><div>+ drawDivision(): void</div><div>+ addPolygon(std::vector&lt;double&gt;&amp;, Channel&amp;, double, int): void</div><div>+ addLissajous(std::vector&lt;double&gt;&amp;, Channel&amp;, std::vector&lt;double&gt;&amp;, Channel&amp;, int, int): void</div><div>+ fitRangeInView(QRectF&amp;): void</div><div># mouseDoubleClickEvent(QMouseEvent*): void</div><div># mousePressEvent(QMouseEvent*): void</div><div># mouseMoveEvent(QMouseEvent*): void</div><div># mouseReleaseEvent(QMouseEvent*): void</div><div># wheelEvent(QWheelEvent*): void</div><div>+ sendTimePos(double): void</div><div>+ sendLevel(double): void</div></div>	

TextItem	QGraphicsItem
<div><div>- mpFont: QFont* {readOnly}</div><div>- mText: QString</div><div>- mPosition: QPointF</div><div>- mBoundingRect: QRectF</div></div>	
<div><div>+ TextItem()</div><div>+ TextItem(QString&amp;, QPointF&amp;, QRectF&amp;)</div><div>+ ~TextItem()</div><div>+ setFont(QFont*): void</div><div>+ setText(QString&amp;, QPointF&amp;): void</div><div># boundingRect(): QRectF {query}</div><div># paint(QPainter*, QStyleOptionGraphicsItem*, QWidget*)</div></div>	

PolyLineItem	QGraphicsItem
<div><div>- mPolyLine: QPolygonF</div><div>- mpPen: QPen* {readOnly}</div></div>	
<div><div>+ PolyLineItem()</div><div>+ ~PolyLineItem()</div><div>+ setPen(QPen*): void</div><div>+ getPolyLine(): QPolygonF&amp;</div><div># boundingRect(): QRectF {query}</div><div># paint(QPainter*, QStyleOptionGraphicsItem*, QWidget*)</div></div>	



<i>em</i>
void

<i>em</i>
void

```
#  mouseMoveEvent(QMouseEvent*): void
#  mouseReleaseEvent(QMouseEvent*): void
+  sendFreqPos(double): void
+  sendLevel(double): void
+  setFilterFunction(int): void
+  setFFT_ValueFunction(int): void
+  setFFT_FloatingAvgValues(int): void
-  updateFrequencyDivision(): void
```

