```
//@version=5
// special credits to olinmck, tradingview and anyone else I missed
indicator(title="NovAlgo - Fast Signals", overlay=true)

// <inputs>

// <inputs - EMAs>
i_ema1Source = input.source(title='1', defval=close, group='EMAs', inline='ema1')
i_ema1Period = input.int(title='', defval=20, group='EMAs', inline='ema1')
i_ema1Color = input.color(title='', defval=color.red, group='EMAs', inline='ema1')
i_ema1Show = input.bool(title='Show?', defval=true, group='EMAs', inline='ema1')
i_ema2Source = input.source(title='2', defval=close, group='EMAs', inline='ema2')
i_ema2Period = input.int(title='', defval=50, group='EMAs', inline='ema2')
i_ema2Color = input.color(title='', defval=color.orange, group='EMAs', inline='ema2')
i_ema2Show = input.bool(title='Show?', defval=true, group='EMAs', inline='ema2')
i_ema3Source = input.source(title='3', defval=close, group='EMAs', inline='ema3')
i_ema3Period = input.int(title='', defval=100, group='EMAs', inline='ema3')
i_ema3Color = input.color(title='', defval=color.green, group='EMAs', inline='ema3')
i_ema3Show = input.bool(title='Show?', defval=true, group='EMAs', inline='ema3')
i_ema4Source = input.source(title='4', defval=close, group='EMAs', inline='ema4')
i_ema4Period = input.int(title='', defval=200, group='EMAs', inline='ema4')
i_ema4Color = input.color(title='', defval=color.white, group='EMAs', inline='ema4')
i_ema4Show = input.bool(title='Show?', defval=true, group='EMAs', inline='ema4')
i_ema5Source = input.source(title='5', defval=close, group='EMAs', inline='ema5')
i_ema5Period = input.int(title='', defval=9, group='EMAs', inline='ema5')
i_ema5Color = input.color(title='', defval=color.yellow, group='EMAs', inline='ema5')
i_ema5Show = input.bool(title='Show?', defval=true, group='EMAs', inline='ema5')
```

```
// <inputs - EMAs>

// <inputs - MA Cloud>
i_source = input.source(title='Source', defval=close, group='MA Cloud')
i_lengthShort = input.int(title='Period - Short', defval=4, group='MA Cloud')
i_lengthLong = input.int(title='Period - Long', defval=20, group='MA Cloud')
i_lengthNA = input.int(title='Period - NA', defval=20, group='MA Cloud')
i_upTrendColor = input.color(title='Uptrend', defval=#77ba7a, group='MA Cloud',
inline='maCloudColor')
i_downTrendColor = input.color(title='Downtrend', defval=#9c3434, group='MA
Cloud', inline='maCloudColor')
i_showCloud = input.bool(title='Show?', defval=true, group='MA Cloud')
// </inputs - MA Cloud>

// <inputs - QQE Signals> - OPTIMIZED FOR FASTER SIGNALS
i_rsiPeriod = input.int(title='RSI Length', defval=8, group='QQE Signals')
i_rsiSmoothPeriod = input.int(title='RSI Smoothing', defval=3, group='QQE
Signals')
i_qqeFactor = input.float(title='Fast QQE Factor', defval=3.2, group='QQE
Signals')
i_qqeSource = input.source(title='Source', defval=close, group='QQE Signals')
i_qqeLongFlagColor = input.color(title='Long Flag', defval=color.green,
group='QQE Signals', inline='qqeColorLong')
i_qqeLongTextColor = input.color(title='Long Text', defval=color.white,
group='QQE Signals', inline='qqeColorLong')
i_qqeShortFlagColor = input.color(title='Short Flag', defval=color.red,
group='QQE Signals', inline='qqeColorShort')
i_qqeShortTextColor = input.color(title='Short Text', defval=color.white,
group='QQE Signals', inline='qqeColorShort')
i_showQqe = input.bool(title='Show?', defval=true, group='QQE Signals')
// </inputs - QQE Signals>

// <inputs - VWAP>
i_hideonDWM = input(false, title="Hide VWAP on 1D or Above", group="VWAP")
var i_anchor = input.string(defval = "Session", title="VWAP - Anchor Period",
options=["Session", "Week", "Month", "Quarter", "Year", "Decade", "Century",
"Earnings", "Dividends", "Splits"], group="VWAP")
i_src = input(title = "Source", defval = hlc3, group="VWAP")
i_offset = input(0, title="Offset", group="VWAP")
i_vwapColor = input.color(#2962FF, title='Color', group="VWAP")
i_showBand1 = input(true, title="", group="Standard Deviation Bands Settings",
inline="band_1")
```

```
i_stdevMult1 = input(1.0, title="Bands Multiplier #1", group="Standard
Deviation Bands Settings", inline="band_1")
i_bandColor1 = input.color(title="", defval=color.green, group="Standard
Deviation Bands Settings", inline="band_1")
i_bandFillColor1 = input.color(title="", defval=color.new(color.green, 95),
group="Standard Deviation Bands Settings", inline="band_1")
i_showBand2 = input(false, title="", group="Standard Deviation Bands Settings",
inline="band_2")
i_stdevMult2 = input(2.0, title="Bands Multiplier #2", group="Standard
Deviation Bands Settings", inline="band_2")
i_bandColor2 = input.color(title="", defval=color.olive, group="Standard
Deviation Bands Settings", inline="band_2")
i_bandFillColor2 = input.color(title="", defval=color.new(color.olive, 95),
group="Standard Deviation Bands Settings", inline="band_2")
i_showBand3 = input(false, title="", group="Standard Deviation Bands Settings",
inline="band_3")
i_stdevMult3 = input(3.0, title="Bands Multiplier #3", group="Standard
Deviation Bands Settings", inline="band_3")
i_bandColor3 = input.color(title="", defval=color.teal, group="Standard
Deviation Bands Settings", inline="band_3")
i_bandFillColor3 = input.color(title="", defval=color.new(color.teal, 95),
group="Standard Deviation Bands Settings", inline="band_3")
i_showVwap = input.bool(title='Show?', defval=true, group='VWAP')
// </inputs - VWAP>

// </inputs>

if barstate.islast and ta.cum(volume) == 0
   runtime.error("No volume is provided by the data vendor.")

// <funcs>

f_display(_show) => _show ? display.all : display.none

// <funcs - MA Cloud>
f_colorTrend(_a, _b, _transp) => _a > _b ? color.new(i_upTrendColor, _transp) :
color.new(i_downTrendColor, _transp)
// </funcs - MA Cloud>

// </funcs>

// <vars>
// <vars - EMAs>
v_ema1 = ta.ema(i_ema1Source, i_ema1Period)
v_ema2 = ta.ema(i_ema2Source, i_ema2Period)
v_ema3 = ta.ema(i_ema3Source, i_ema3Period)
v_ema4 = ta.ema(i_ema4Source, i_ema4Period)
v_ema5 = ta.ema(i_ema5Source, i_ema5Period)
// </vars - EMAs>

// <vars - MA Cloud>
v_emaShort = ta.ema(i_source, i_lengthShort)
v_emaLong = ta.ema(i_source, i_lengthLong)
v_sma = ta.sma(i_source, i_lengthNA)
// </vars - MA Cloud>

// <vars - QQE Signals> - OPTIMIZED FOR FASTER RESPONSE
v_longBand = 0.0
v_shortBand = 0.0
v_trend = 0
v_qqeXLong = 0
v_qqeXShort = 0
v_wildersPeriod = i_rsiPeriod * 2 - 1
v_rsi = ta.rsi(i_qqeSource, i_rsiPeriod)
v_rsiMa = ta.ema(v_rsi, i_rsiSmoothPeriod)
v_atrRsi = math.abs(v_rsiMa[1] - v_rsiMa)
v_maAtrRsi = ta.ema(v_atrRsi, v_wildersPeriod)
v_deltaFastAtrRsi = ta.ema(v_maAtrRsi, v_wildersPeriod) * i_qqeFactor
v_newShortBand = v_rsiMa + v_deltaFastAtrRsi
v_newLongBand = v_rsiMa - v_deltaFastAtrRsi
v_longBand := v_rsiMa[1] > v_longBand[1] and v_rsiMa > v_longBand[1] ?
math.max(v_longBand[1], v_newLongBand) : v_newLongBand
v_shortBand := v_rsiMa[1] < v_shortBand[1] and v_rsiMa < v_shortBand[1] ?
math.min(v_shortBand[1], v_newShortBand) : v_newShortBand
v_cross1 = ta.cross(v_longBand[1], v_rsiMa)
v_trend := ta.cross(v_rsiMa, v_shortBand[1]) ? 1 : v_cross1 ? -1 :
nz(v_trend[1], 1)
v_fastAtrRsiTL = v_trend == 1 ? v_longBand : v_shortBand

// find all the QQE Crosses - OPTIMIZED
v_qqeXLong := nz(v_qqeXLong[1])
v_qqeXShort := nz(v_qqeXShort[1])
v_qqeXLong := v_fastAtrRsiTL < v_rsiMa ? v_qqeXLong + 1 : 0
```

```
v_qqeXShort := v_fastAtrRsiTL > v_rsiMa ? v_qqeXShort + 1 : 0

// conditions - WITH ANTI-REPAINTING PROTECTION
// Only trigger signals on confirmed closed bars to prevent repainting
v_qqeLong = v_qqeXLong[1] == 1 and barstate.isconfirmed ? v_fastAtrRsiTL[1] -
50 : na
v_qqeShort = v_qqeXShort[1] == 1 and barstate.isconfirmed ? v_fastAtrRsiTL[1] -
50 : na
// </vars - QQE Signals>

// <vars - VWAP>
v_newEarnings = request.earnings(syminfo.tickerid, earnings.actual,
barmerge.gaps_on, barmerge.lookahead_on, ignore_invalid_symbol=true)
v_newDividends = request.dividends(syminfo.tickerid, dividends.gross,
barmerge.gaps_on, barmerge.lookahead_on, ignore_invalid_symbol=true)
v_newSplit = request.splits(syminfo.tickerid, splits.denominator,
barmerge.gaps_on, barmerge.lookahead_on, ignore_invalid_symbol=true)

v_isNewPeriod = switch i_anchor
   "Earnings"  => not na(v_newEarnings)
   "Dividends" => not na(v_newDividends)
   "Splits"    => not na(v_newSplit)
   "Session"   => timeframe.change("D")
   "Week"      => timeframe.change("W")
   "Month"     => timeframe.change("M")
   "Quarter"   => timeframe.change("3M")
   "Year"      => timeframe.change("12M")
   "Decade"    => timeframe.change("12M") and year % 10 == 0
   "Century"   => timeframe.change("12M") and year % 100 == 0
   => false

v_isEsdAnchor = i_anchor == "Earnings" or i_anchor == "Dividends" or i_anchor
== "Splits"
if na(i_src[1]) and not v_isEsdAnchor
   v_isNewPeriod := true

float v_vwapValue = na
float v_upperBandValue1 = na
float v_lowerBandValue1 = na
float v_upperBandValue2 = na
float v_lowerBandValue2 = na
float v_upperBandValue3 = na
float v_lowerBandValue3 = na

if not (i_hideonDWM and timeframe.isdwm)
   [_vwap, _stdevUpper, _] = ta.vwap(i_src, v_isNewPeriod, 1)
   v_vwapValue := _vwap
   v_stdevAbs = _stdevUpper - _vwap
   v_upperBandValue1 := _vwap + v_stdevAbs * i_stdevMult1
   v_lowerBandValue1 := _vwap - v_stdevAbs * i_stdevMult1
   v_upperBandValue2 := _vwap + v_stdevAbs * i_stdevMult2
   v_lowerBandValue2 := _vwap - v_stdevAbs * i_stdevMult2
   v_upperBandValue3 := _vwap + v_stdevAbs * i_stdevMult3
   v_lowerBandValue3 := _vwap - v_stdevAbs * i_stdevMult3
// </vars - VWAP>

// </vars>

// <plots>

// <plots - EMAs>
plot(v_ema1, title='EMA1', color=i_ema1Color, display=f_display(i_ema1Show))
plot(v_ema2, title='EMA2', color=i_ema2Color, display=f_display(i_ema2Show))
plot(v_ema3, title='EMA3', color=i_ema3Color, display=f_display(i_ema3Show))
plot(v_ema4, title='EMA4', color=i_ema4Color, display=f_display(i_ema4Show))
plot(v_ema5, title='EMA5', color=i_ema5Color, display=f_display(i_ema5Show))
// </plots - EMAs>

// <plots - MA Cloud>
p_short = plot(v_emaShort, title='Short', color=f_colorTrend(v_emaShort,
v_emaLong, 30), display=f_display(i_showCloud))
p_long = plot(v_emaLong, title='L', color=f_colorTrend(v_emaShort, v_emaLong,
30), display=f_display(i_showCloud))
p_reg = plot(v_sma, title='K', color=f_colorTrend(v_emaLong, v_sma, 70),
display=f_display(i_showCloud))

fill(p_short, p_long, color=f_colorTrend(v_emaShort, v_emaLong, 30),
display=f_display(i_showCloud))
fill(p_reg, p_long, color=f_colorTrend(v_emaLong, v_sma, 70),
display=f_display(i_showCloud))
// </plots - MA Cloud>

// <plots - QQE Signals>
```

```
plotshape(v_qqeLong, title='QQE long', text='Long',
textcolor=i_qqeLongTextColor, style=shape.labelup, location=location.belowbar,
color=i_qqeLongFlagColor, size=size.tiny, display=f_display(i_showQqe))
plotshape(v_qqeShort, title='QQE short', text='Short',
textcolor=i_qqeShortTextColor, style=shape.labeldown,
location=location.abovebar, color=i_qqeShortFlagColor, size=size.tiny,
display=f_display(i_showQqe))
// </plots - QQE Signals>

// <plots - VWAP>
plot(v_vwapValue, title="VWAP", color=i_vwapColor, offset=i_offset, display =
f_display(i_showVwap))

p_upperBand1 = plot(v_upperBandValue1, title="Upper Band #1",
color=i_bandColor1, offset=i_offset, display = f_display(i_showVwap and
i_showBand1))
p_lowerBand1 = plot(v_lowerBandValue1, title="Lower Band #1",
color=i_bandColor1, offset=i_offset, display = f_display(i_showVwap and
i_showBand1))
fill(p_upperBand1, p_lowerBand1, title="Bands Fill #1", color=
i_bandFillColor1, display = f_display(i_showVwap and i_showBand1))

p_upperBand2 = plot(v_upperBandValue2, title="Upper Band #2",
color=i_bandColor2, offset=i_offset, display = f_display(i_showVwap and
i_showBand2))
p_lowerBand2 = plot(v_lowerBandValue2, title="Lower Band #2",
color=i_bandColor2, offset=i_offset, display = f_display(i_showVwap and
i_showBand2))
fill(p_upperBand2, p_lowerBand2, title="Bands Fill #2", color=
i_bandFillColor2, display = f_display(i_showVwap and i_showBand2))

p_upperBand3 = plot(v_upperBandValue3, title="Upper Band #3",
color=i_bandColor3, offset=i_offset, display = f_display(i_showVwap and
i_showBand3))
p_lowerBand3 = plot(v_lowerBandValue3, title="Lower Band #3",
color=i_bandColor3, offset=i_offset, display = f_display(i_showVwap and
i_showBand3))
fill(p_upperBand3, p_lowerBand3, title="Bands Fill #3", color=
i_bandFillColor3, display = f_display(i_showVwap and i_showBand3))
// </plots - VWAP>

// </plots>
// <alerts>

// <alerts - QQE Signals with Webhook JSON>
alertcondition(not na(v_qqeLong), title='QQE Long Signal',
    message='{"action": "BUY", "symbol": "{{ticker}}", "secret":
"my_secret_key_123"}')

alertcondition(not na(v_qqeShort), title='QQE Short Signal',
    message='{"action": "SELL", "symbol": "{{ticker}}", "secret":
"my_secret_key_123"}')
// </alerts - QQE Signals>

// </alerts>
```