// This source code is subject to the terms of the Mozilla Public License 2.0 at https://mozilla.org/MPL/2.0/

// Â© RajeshMoriyani

//This is all combined Indicator for Bootsie's Boom & Crash Strategy

//The code for following indicators/parts is inspired by/borrowed from : (and hence, the credits & copyrights(if any) for them remain with their respective original authors)

//(1) TSI part : TSI with histogram and MA - SamX

//(2) WAE part : Waddah Attar Explosion V2 [SHK]

//@version=5

indicator("RSI+EMAs Ribbon+MA+TSI+WAE for Bootsie's Boom & Crash Strat", shorttitle="Bootsie's Boom & Crash (RSI+EMAs Ribbon+MA+TSI+WAE)")

RSI\_SETTINGS = "RSI Settings"

rsiLengthInput = input.int(500, minval=1, title="RSI Length", group=RSI\_SETTINGS)

rsiSourceInput = input.source(close, title="Source for RSI", group=RSI\_SETTINGS)

MA10\_SETTINGS = "MA-10 Settings"

ma10SourceInput = input.source(close, title="Source for MA-10", group=MA10\_SETTINGS)

maLengthInput = input.int(10, title="MA-10", group=MA10\_SETTINGS)

EMA\_SETTINGS = "EMA Settings"

maSourceInput = input.source(hlcc4, title="Source for EMAs", group=EMA\_SETTINGS)

maLengthInput1 = input.int(1, title="EMA1", group=EMA\_SETTINGS)

maLengthInput2 = input.int(2, title="EMA2", group=EMA\_SETTINGS)

maLengthInput3 = input.int(3, title="EMA3", group=EMA\_SETTINGS)

maLengthInput4 = input.int(4, title="EMA4", group=EMA\_SETTINGS)

maLengthInput5 = input.int(5, title="EMA5", group=EMA\_SETTINGS)

PRICE\_PIVOTS = "Price High Low Pivots"

lookbackLength = input.int(150, "Lookback Period", minval = 10, maxval = 500, group = PRICE\_PIVOTS) // max bars to lookback

H\_LINES = "Horizontal Lines/Levels"

h\_line\_color = input(color.new(color.white, 70), "", inline="HLines", group=H\_LINES)

hLine1 = input(85, "Level-1", inline="HLines1", group=H\_LINES)

hLine2 = input(0, "Level-2", inline="HLines2", group=H\_LINES)

hLine3 = input(0, "Level-3", inline="HLines3", group=H\_LINES)

hLine4 = input(0, "Level-4", inline="HLines4", group=H\_LINES)

hLine5 = input(10, "Level-5", inline="HLines5", group=H\_LINES)

//TSI :

TSI\_SETTINGS = 'TSI Settings'

showTSIAsTrendBar = input.bool(title='Show TSI as Trend Bar ', defval=true, group=TSI\_SETTINGS, tooltip='Show/Hide TSI as Trend Bar at the top.\n\nGreen/Red Bars when TSI above/below Signal Line.\nUp/Down Triangles when TSI crosses MA')

long = input.int(title='Long-Preiod Length', defval=13, group=TSI\_SETTINGS, tooltip='Original default: 25')

short = input.int(title='Short-Period Length', defval=6, group=TSI\_SETTINGS, tooltip='Original default: 13')

signal = input.int(title='Signal Preiod Length', defval=4, group=TSI\_SETTINGS, tooltip='Original default: 13')

priceSource = input.source(title='Source for price', defval=close, group=TSI\_SETTINGS, tooltip='Source for price information. Default: Close')

TSI\_MA\_SETTINGS = "TSI Moving Average Settings"

maType = input.string(title='TSI Moving Average Calculation', group=TSI\_MA\_SETTINGS, options=['Exponential', 'Simple', 'Smoothed', 'Weighted', 'Linear', 'Hull', 'Arnaud Legoux'], defval='Exponential', tooltip='Type of moving average calculation to use (default is Exponential (EMA)). \n\n' +

'Only relevant when "Show floating TSI Moving Average" above is enabled.')

maLength = input.int(title="TSI MA Length", defval=50, minval=1, step=1, group=TSI\_MA\_SETTINGS, tooltip="Number of historical TSI values to use for calculating the MA value. Lower numbers will react more quickly, while higher numbers will better depict longer trends.")

TSI\_ALMA\_SETTINGS = 'TSI ALMA Additional Settings'

almaSigma = input.float(title="Sigma", defval=6, minval=0, maxval=100, step=0.1, group=TSI\_ALMA\_SETTINGS, tooltip='Standard deviation applied to the MA. Higher values tend to make the line smoother. Default: 6')

almaOffset = input.float(title="Offset", defval=0.85, minval=0, maxval=1, step=0.01, group=TSI\_ALMA\_SETTINGS, tooltip='Standard deviation applied to the MA. Higher values tend to make the line smoother, while lower values make it more responsive. Default: 0.85')

// Define the colors for the TSI so we can re-use them in the fill zone as well...

tsiBullColor = color.rgb(0, 255, 0, 0)

tsiBearColor = color.rgb(255, 0, 0, 0)

//WAE :

WAE\_SETTINGS = 'WAE (Waddah Attar Explosion) Settings'

showWAEAsTrendBar = input.bool(title= 'Show WAE as Trend Bar ', defval= true, group= WAE\_SETTINGS, tooltip='Show/Hide WAE as Trend Bar at the bottom.\n\nDEAD Zone: White Bars (Low Volatility).\nUptrend: Green/Lime for rising/falling Volatility.\nDowntrend: Red/Orange for rising/falling Volatility.')

showWAECrossoversEL = input.bool(title= 'Show WAE-EL Crossovers', defval= false, group= WAE\_SETTINGS, tooltip='The checkbox will enable/disable showing WAE crossovers with EL(Explosion Line).\n\nUp/Down Triangles when WAE crosses EL.')

sensitivity = input.int(title="Sensitivity", defval=150, minval=1, maxval=9999, step=1, group= WAE\_SETTINGS)

fastLength= input.int(title="FastEMA Length", defval=20, minval=1, maxval=999, step=1, group= WAE\_SETTINGS)

slowLength= input.int(title="SlowEMA Length", defval=40, minval=1, maxval=999, step=1, group= WAE\_SETTINGS)

channelLength= input.int(title="BB Channel Length", defval=20, minval=1, maxval=999, step=1, group= WAE\_SETTINGS)

mult= input.float(title="BB Stdev Multiplier", defval=2.0, minval=0, maxval=100, step=0.1, group= WAE\_SETTINGS)

// Function to calculate smoothed moving average...

smoothedMovingAvg(src, len) =>

smma = 0.0

// TV will complain about the use of the ta.sma function use inside a function saying that it should be called on each calculation,

// but since we're only using it once to set the initial value for the smoothed MA (when the previous smma value is NaN - Not a Number)

// and using the previous smma value for each subsequent iteration, this can be safely ignored

smma := na(smma[1]) ? ta.sma(src, len) : (smma[1] \* (len - 1) + src) / len

smma

double\_smooth(src, long, short) =>

fist\_smooth = ta.ema(src, long)

ta.ema(fist\_smooth, short)

//WAE :

calc\_macd(source, fastLength, slowLength) =>

fastMA = ta.ema(source, fastLength)

slowMA = ta.ema(source, slowLength)

fastMA - slowMA

calc\_BBUpper(source, length, mult) =>

basis = ta.sma(source, length)

dev = mult \* ta.stdev(source, length)

basis + dev

calc\_BBLower(source, length, mult) =>

basis = ta.sma(source, length)

dev = mult \* ta.stdev(source, length)

basis - dev

//Calculations

//RSI

rsi = ta.rsi(rsiSourceInput, rsiLengthInput)

//Hi Lo Pivots for finding Price Range

pivotHigh = ta.highest(maSourceInput, lookbackLength)

pivotLow = ta.lowest(maSourceInput , lookbackLength)

var scaleLow = 0.0

var scaleHigh = 100.0

scaledValue = scaleLow + (maSourceInput - pivotLow) \* (scaleHigh - scaleLow) / (pivotHigh - pivotLow)

//EMAs

ma1 = ta.ema(scaledValue, maLengthInput1)

ma2 = ta.ema(scaledValue, maLengthInput2)

ma3 = ta.ema(scaledValue, maLengthInput3)

ma4 = ta.ema(scaledValue, maLengthInput4)

ma5 = ta.ema(scaledValue, maLengthInput5)

//MA-10

scaledValueMA10 = scaleLow + (ma10SourceInput - pivotLow) \* (scaleHigh - scaleLow) / (pivotHigh - pivotLow)

ma10 = ta.sma(scaledValueMA10, maLengthInput)

//TSI

price = priceSource

pc = ta.change(price)

double\_smoothed\_pc = double\_smooth(pc, long, short)

double\_smoothed\_abs\_pc = double\_smooth(math.abs(pc), long, short)

tsi\_value = 100 \* (double\_smoothed\_pc / double\_smoothed\_abs\_pc)

signal\_line = ta.ema(tsi\_value, signal)

//TSI Fill

// Identify if TSI is above or below signal line

tsiBullish = tsi\_value > signal\_line

tsiBearish = tsi\_value < signal\_line

tsiTrendColor = tsiBullish ? color.new(tsiBullColor, 63) : tsiBearish ? color.new(tsiBearColor, 63) : na

// TSI MA fun

floatingMA = maType == 'Exponential' ? ta.ema(tsi\_value, maLength) : maType == 'Simple' ? ta.sma(tsi\_value, maLength) : maType == 'Weighted' ? ta.wma(tsi\_value, maLength) : maType == 'Linear' ? ta.linreg(tsi\_value, maLength, 0) : maType == 'Hull' ? ta.hma(tsi\_value, maLength) : maType == 'Smoothed' ? smoothedMovingAvg(tsi\_value, maLength) : maType == 'Arnaud Legoux' ? ta.alma(tsi\_value, maLength, almaOffset, almaSigma) : na

tsiAboveMA = ta.crossover(tsi\_value, floatingMA)

tsiBelowMA = ta.crossunder(tsi\_value, floatingMA)

//WAE :

DEAD\_ZONE = nz(ta.rma(ta.tr(true), 100)) \* 3.7

t1 = (calc\_macd(close, fastLength, slowLength) - calc\_macd(close[1], fastLength, slowLength))\*sensitivity

e1 = (calc\_BBUpper(close, channelLength, mult) - calc\_BBLower(close, channelLength, mult))

t1Abs = math.abs(t1)

waeAboveEL = ta.crossover(t1Abs, e1)

waeBelowEL = ta.crossunder(t1Abs, e1)

waeNotInDeadZone = t1Abs > DEAD\_ZONE

waeTrendFalling = t1Abs < t1Abs[1]

waeTrendColour = waeNotInDeadZone ? ((t1 >= 0) ? (waeTrendFalling ? color.lime : color.green)

: (waeTrendFalling ? color.orange : color.red))

: color.white

//Plots :

//RSI

plot(rsi,"RSI", color= color.yellow, linewidth=2)

//EMAs Ribbon

plot(ma1, "Scaled EMA", color= color.lime)

plot(ma2, "Scaled EMA2", color= color.teal)

plot(ma3, "Scaled EMA3", color= color.orange)

plot(ma4, "Scaled EMA4", color= color.fuchsia)

plot(ma5, "Scaled EMA5", color= color.red)

//MA-10

plot(ma10, "Scaled MA-10", color= color.white, linewidth=2)

//H Lines

line1 = hline((hLine1 > 0) ? hLine1 : na, "", color=(hLine1 > 0) ? h\_line\_color : na, linestyle=hline.style\_dashed)

line2 = hline((hLine2 > 0) ? hLine2 : na, "", color=(hLine2 > 0) ? h\_line\_color : na, linestyle=hline.style\_dashed)

line3 = hline((hLine3 > 0) ? hLine3 : na, "", color=(hLine3 > 0) ? h\_line\_color : na, linestyle=hline.style\_dashed)

line4 = hline((hLine4 > 0) ? hLine4 : na, "", color=(hLine4 > 0) ? h\_line\_color : na, linestyle=hline.style\_dashed)

line5 = hline((hLine5 > 0) ? hLine5 : na, "", color=(hLine5 > 0) ? h\_line\_color : na, linestyle=hline.style\_dashed)

//TSI : show only as Trend

plotshape(showTSIAsTrendBar and tsiBullish ? tsi\_value : na , title= "TSI Up Trend", location= location.top, color= tsiTrendColor, style= shape.square)

plotshape(showTSIAsTrendBar and tsiBearish ? tsi\_value : na, title= "TSI Down Trend", location= location.top, color= tsiTrendColor, style= shape.square)

//TSI crossovers with MA

plotshape(showTSIAsTrendBar and tsiAboveMA ? tsiAboveMA : na , title= "TSI above MA", location= location.top, color= color.yellow, size= size.tiny, style= shape.triangleup)

plotshape(showTSIAsTrendBar and tsiBelowMA ? tsiBelowMA : na , title= "TSI below MA", location= location.top, color= color.yellow, size= size.tiny, style= shape.triangledown)

//WAE : show only as Trend

plotshape(showWAEAsTrendBar ? t1Abs : na , title= "WAE Trend Bar", location= location.bottom, color= color.new(waeTrendColour, 35), size= size.tiny, style= shape.square)

//WAE crossovers with Explosion Line

plotshape(showWAEAsTrendBar and showWAECrossoversEL and waeAboveEL ? t1Abs : na , title= "WAE above EL", location= location.bottom, color= color.yellow, size= size.tiny, style= shape.triangleup)

plotshape(showWAEAsTrendBar and showWAECrossoversEL and waeBelowEL ? t1Abs : na , title= "WAE below EL", location= location.bottom, color= color.yellow, size= size.tiny, style= shape.triangledown)