//**@version=**5

indicator("Supertrended RSI\_modified", overlay = false)

// Input parameters

relativeStrengthIndexLength = input.int(14, title="RSI Length", group="RSI Settings")

smoothingLength = input.int(21, title="RSI Smoothing Length", group="RSI Settings")

rsiInputSource = input.source(close, title="RSI Source", group="RSI Settings")

upperlimit = input.int(70, title="RSI upper limit", group="RSI Settings")

lowerlimit = input.int(30, title="RSI lower limit", group="RSI Settings")

middleLine = input.int(50, title="RSI middle line",minval = 20,maxval=90, group="RSI Settings")

middleLine\_offset = input.int(0, title="RSI middle line offset", minval = 0,maxval=10, group="RSI Settings")

trendFactor = input.float(0.8, title="Factor", group="Super Trend Settings")

averageTrueRangeLength = input.int(10, title="ATR Length", group="Super Trend Settings")

// Input for account information  value

StLotSize=input.float(title=" lot size", defval=0.5,minval=0.2 ,maxval=2.0 , group="trading parameters")

accountValue = input.float(1500, title="Account Value", group="trading parameters")

Trigger\_offset = input.int(0, title="Trigger\_Offset",minval = 0,maxval=5, group="trading parameters")

colorUp = input.color(#00ffbb, title="Up Color")

colorDown = input.color(#ff1100, title="Down Color")

//hechnatchi code start

REAL = input(false, title="Use Real Close?", group="Heikin Ashi [Improved]", tooltip="Get the real closing price per bar close")

HA = input(true, title="Add Original Heikin Ashi Smoothness?", group="Heikin Ashi [Improved]", tooltip="Add an extra layer of smoothness.")

s = input.int(1, minval=1, maxval=100, title="Candle Smoothness", group="Heikin Ashi [Improved]")

///////////////////////////////////////////////////////////////////////////////// Heikin Ashi [improved]

close0 =  (open + high + low + close) / 4

close0 :=  na(close0)? nz(na) : (nz(close0) + nz(close)) / 2

open0 = float(na)

open0 :=  na(open0[1])? nz(na) :  (nz(open0[s]) + nz(close0[1])) / 2

high0 = math.max(high, math.max(open0,close0))

low0 = math.min(low, math.min(open0,close0))

h = math.max(high0,low0)

l = math.min(high0,low0)

///////////////////////////////////////////////////////////////////////////////// Original Heikin Ashi Smoothness

haClose = (open0 + h + l + close0) / 4

haOpen = float(na)

haOpen := na(haOpen[1]) ? (open0 + close0) / 2 : (nz(haOpen[1]) + nz(haClose[1])) / 2

haHigh = math.max(h, math.max(haOpen, haClose))

haLow = math.min(l, math.min(haOpen, haClose))

//plotcandle(haOpen,haHigh,haLow,close, color=haClose>haOpen?color.lime:color.red, title="Heikin Ashi [improved] + Original Heikin Ashi Smoothness")

///////////////////////////////////////////////////////////////////////////////// Candle Mode

o\_ = HA? haOpen : open0

h\_ = HA? haHigh : h

l\_ = HA? haLow : l

c\_ = HA? REAL? close: haClose : REAL? close: close0

//hechnatchi code end

Uoffset=middleLine + middleLine\_offset

Doffset=middleLine - middleLine\_offset

//read symbol

var symbol = syminfo.ticker //get currency symbol

//modify for gold and silver

if (symbol == "XAUUSD" )

    StLotSize:=0.1

else if(symbol == "XAGUSD")

    StLotSize:=0.3

// Function to calculate ATR

calculateATR(source, atrLength) =>

    highestHigh = ta.highest(source, atrLength)

    lowestLow = ta.lowest(source, atrLength)

    trueRange = na(highestHigh[1]) ? highestHigh - lowestLow : math.max(highestHigh - lowestLow, math.abs(highestHigh - source[1]), math.abs(lowestLow - source[1]))

    ta.rma(trueRange, atrLength)

// Function to calculate Supertrend

calculateSupertrend(factor, atrPeriod, source) =>

    priceSource = source

    atr = calculateATR(source, atrPeriod)

    upperBand = priceSource + factor \* atr

    lowerBand = priceSource - factor \* atr

    prevLowerBand = nz(lowerBand[1])

    prevUpperBand = nz(upperBand[1])

    lowerBand := lowerBand > prevLowerBand or source[1] < prevLowerBand ? lowerBand : prevLowerBand

    upperBand := upperBand < prevUpperBand or source[1] > prevUpperBand ? upperBand : prevUpperBand

**int** trendDirection = na

**float** supertrendValue = na

    prevSupertrend = supertrendValue[1]

    if na(atr[1])

        trendDirection := 1

    else if prevSupertrend == prevUpperBand

        trendDirection := source > upperBand ? -1 : 1

    else

        trendDirection := source < lowerBand ? 1 : -1

    supertrendValue := trendDirection == -1 ? lowerBand : upperBand

    [supertrendValue, trendDirection]

// Calculating RSI

//rsiValue =   ta.rsi(rsiInputSource, relativeStrengthIndexLength)

rsiValue =   ta.rsi(c\_, relativeStrengthIndexLength)

// Calculating Supertrend based on RSI values

[rsiSupertrend, trendDirection] = calculateSupertrend(trendFactor, averageTrueRangeLength, rsiValue)

// Plotting

supertrendPlot = plot(rsiSupertrend, color = trendDirection == -1 ? colorUp : colorDown, title="Supertrend", linewidth = 2)

rsiPlot = plot(rsiValue, color = color.blue, title="RSI", linewidth = 2)

overboughtLine = plot(upperlimit, color = color.white, title="Overbought Line", linewidth = 1)

oversoldLine = plot(lowerlimit, color = color.white, title="Oversold Line", linewidth = 1)

midindline=plot(middleLine, color = color.white, title="Mid Line", linewidth = 1)

OffsetUp=plot(Uoffset, color = color.orange, title="up offset", linewidth = 1)

OffsetDown=plot(Doffset, color = color.orange, title="downoffset", linewidth = 1)

var **bool** buy=false

var **bool** sell=false

var **bool** triggerbuy=false

var **bool** triggersell=false

var **bool** triggerclosebuy=false

var **bool** triggerclosesell=false

var **bool** EarlyTriggerCloseBuy=false

var **bool** EarlyTriggerCloseSell=false

if  (ta.crossover(rsiSupertrend[Trigger\_offset], rsiValue[Trigger\_offset]) and trendDirection ==1 and not sell)

    label.new(bar\_index,close,"",yloc=yloc.abovebar,style=label.style\_label\_down,color = color.red,size=size.small,force\_overlay = true)

    if(buy)

        label.new(bar\_index,close,"close buy",yloc=yloc.abovebar,style=label.style\_cross,color = color.yellow,size=size.small,force\_overlay = true,textcolor = color.white)

        EarlyTriggerCloseBuy:=true

    else

        EarlyTriggerCloseBuy:=false

    sell:=true

    buy:=false

    triggersell:=true

else if (ta.crossunder(rsiSupertrend[Trigger\_offset], rsiValue[Trigger\_offset]) and trendDirection ==-1 and not buy)

    label.new(bar\_index,close,"",yloc=yloc.belowbar,style=label.style\_label\_up,color = color.green,size=size.small,force\_overlay = true)

    if(sell)

        label.new(bar\_index,close,"close sell",yloc=yloc.belowbar,style=label.style\_flag,color = color.purple,size=size.small,force\_overlay = true,textcolor = color.white)

        EarlyTriggerCloseSell:=true

    else

        EarlyTriggerCloseSell:=false

    buy:=true

    sell:=false

    triggerbuy:=true

var **bool**    closeSellRev=false

var **bool** closeBuyRev=false

if( (not EarlyTriggerCloseSell) and (not EarlyTriggerCloseBuy) )

    closeBuyRev:=ta.crossunder(rsiValue, rsiSupertrend) or ta.crossunder(rsiValue,upperlimit) or ta.crossunder(rsiValue,middleLine)

    closeSellRev:=ta.crossover(rsiValue, rsiSupertrend) or ta.crossover(rsiValue,lowerlimit) or ta.crossover(rsiValue,middleLine)

else if(EarlyTriggerCloseSell)

    sell:=true

    closeSellRev:=true

    closeBuyRev:=false

    EarlyTriggerCloseSell:=false

else if(EarlyTriggerCloseBuy)

    buy:=true

    closeSellRev:=false

    closeBuyRev:=true

    EarlyTriggerCloseBuy:=false

else

    EarlyTriggerCloseBuy:=false

    EarlyTriggerCloseSell:=false

if (closeBuyRev and buy)

    buy:=false

    label.new(bar\_index,close,"close buy",yloc=yloc.abovebar,style=label.style\_cross,color = color.yellow,size=size.small,force\_overlay = true,textcolor = color.white)

    closeBuyRev:=false

    triggerclosebuy:=true

else if(closeSellRev and sell)

    sell:=false

    label.new(bar\_index,close,"close sell",yloc=yloc.belowbar,style=label.style\_flag,color = color.purple,size=size.small,force\_overlay = true,textcolor = color.white)

    closeSellRev:=false

    triggerclosesell:=true

else

    closeSellRev:=false

    triggerclosesell:=false

    closeBuyRev:=false

    triggerclosebuy:=false

//Alerts

alertcondition(triggerbuy, "buy signal triggered")

alertcondition(triggersell, "Sell signal triggered")

alertcondition(triggerclosebuy, "close buy order")

alertcondition(triggerclosesell, "close sell order")