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

// © KioseffTrading

//**@version=**5

indicator("Trended CVD modified", overlay =true, max\_labels\_count = 500, max\_lines\_count = 500, max\_boxes\_count = 500, max\_polylines\_count = 100)

import TradingView/ZigZag/7 as ZigZagLib

wid     = input.int (defval = 1, minval = 1, title = "CVD Line Width"  , group = "CVD Settings")

style   = input.string(defval = "Dotted", title = "CVD Line Style"     , group = "CVD Settings", options = ["Dotted", "Dashed", "Solid"])

upCol   = input.color(defval = #14D990, title = "+ CVD Color"        , group = "CVD Settings")

dnCol   = input.color(defval = #F24968, title = "- CVD Color"        , group = "CVD Settings")

var sty = switch style

    "Dotted" => line.style\_dotted

    "Dashed" => line.style\_dashed

    "Solid"  => line.style\_solid

// Create Zig Zag instance from user settings.

var zigZag = ZigZagLib.newInstance(

  ZigZagLib.Settings.new(

      input.float (0.00001, "Price deviation for reversals (%)", 0.00001, 100.0, 0.5, "0.00001 - 100", group = "Zig Zag Settings"),

      input.int   (50, "Pivot legs", 2, group = "Zig Zag Settings"),

      input       (color.new(#6929F2, 90), "Line color", group = "Zig Zag Settings"),

      input       (false, "Extend to last bar", group = "Zig Zag Settings"),

      input       (false, "Display reversal price", group = "Zig Zag Settings"),

      input       (false, "Display cumulative volume", group = "Zig Zag Settings"),

      input       (false, "Display reversal price change", inline = "priceRev", group = "Zig Zag Settings"),

      input.string("Absolute", "", ["Absolute", "Percent"], inline = "priceRev", group = "Zig Zag Settings"),

      true)

 )

zigZag.update()

type **zigZagData**

**int**                  start

**int**                  end

**array**  <int>         indexArr

**array**  <float>       volArr

**array**  <chart.point> cvdCoords

**float**                vol

**int**                  index

**float**                max = 0

**float**                min = 1e8

var zzd = zigZagData.new(0, 0, indexArr = array.new\_int(), volArr = array.new\_float())

volumeConfig() =>

    switch math.sign(close - open)

        -1 => -volume

        1  =>  volume

        =>     0

method updateData(**array**<**float**> id, **float** value) =>

    id.unshift(value)

zzd.indexArr.updateData(time)

zzd.volArr  .updateData (volumeConfig())

zzd.max     := math.max(zzd.max, high)

zzd.min     := math.min(zzd.min, low)

method drawHistoricalCVD(**array**<**float**> id) =>

    atrZ = ta.atr(14) \* 2

    if line.all.size() != line.all.size()[1]

        if line.all.size() > 2

            getX1     = line.all.last().get\_x1(),    getX2 = line.all.get(line.all.size() - 2).get\_x1()

            if getX2 != zzd.start and getX1 != zzd.end

                zzd.start := getX2, zzd.end   := getX1

                zzd.max   := 0    , zzd.min   := 1e8

                getY1      = line.all.last().get\_y1()

                getY2      = line.all.get(line.all.size() - 2).get\_y1()

                priceRange = getY1 - getY2

                startIndex = zzd.indexArr.indexof(zzd.start)

                endIndex   = zzd.indexArr.indexof(zzd.end)

                slicedVol  = id.slice(endIndex,  startIndex + 1)

                slicedTime = zzd.indexArr.slice(endIndex, startIndex + 1)

                direction  = atrZ \* math.sign(priceRange)

                signDir = math.sign(slicedVol.sum())

                [col, beginTxt, lineCol] = switch signDir

                    -1 => [color.new(dnCol, 50), "" , dnCol]

                    1  => [color.new(upCol, 50), "+", upCol]

                addDivTxt = ""

                sign = math.sign(slicedVol.sum())

                if getY1 < getY2 and sign > 0 or getY1 > getY2 and sign < 0

                    addDivTxt := "\nDivergence"

                label.new(zzd.end, line.all.last().get\_y1() + direction, beginTxt + str.tostring(slicedVol.sum(), format.volume) + addDivTxt, xloc = xloc.bar\_time,

                     style = label.style\_none, textcolor = col,  size = size.small)

                cvd = zigZagData.new(cvdCoords = array.new<**chart.point**>())

                normVol = slicedVol.copy()

                if normVol.size() > 1

                    for i = normVol.size() - 2 to 0

                        normVol.set(i, normVol.get(i + 1) + normVol.get(i))

                    lowDelta = normVol.min(), deltaRange = normVol.range(), priceMin = math.min(getY1, getY2)

                    absRange = math.abs(priceRange), upper = priceMin + absRange \* .75, lower = priceMin + absRange \* .25

                    for i = normVol.size() - 1 to 0

                        normed = lower + ((normVol.get(i) - lowDelta) \* (upper - lower)) / deltaRange

                        cvd.cvdCoords.push(chart.point.from\_time(slicedTime.get(i), normed))

                    polyline.new(cvd.cvdCoords, line\_color = lineCol, xloc = xloc.bar\_time, line\_style = sty,

                          curved = true, line\_width = wid)

method drawLiveCVD(**array**<**float**> id) =>

    atrZ= ta.atr(14) \* 2

    if barstate.islast

        var **polyline** livePoly  = na

        var **label**    liveLabel = na

        getX1      = line.all.last().get\_x1()

        getX2      = line.all.last().get\_x2()

        getY1      = line.all.last().get\_y1()

        getY2      = line.all.last().get\_y2()

        startIndex = zzd.indexArr.indexof(getX1)

        priceRange = math.max(getY1, getY2, zzd.max) - math.min(getY1, getY2, zzd.min)

        slicedVol = id  .slice(0,  startIndex + 1)

        direction = atrZ \* math.sign(priceRange)

        signDir = math.sign(slicedVol.sum())

        [col, beginTxt, lineCol] = switch signDir

            -1 => [color.new(dnCol, 50), "" , dnCol]

            1  => [color.new(upCol, 50), "+", upCol]

        if na(liveLabel)

            liveLabel := label.new(bar\_index + 5, line.all.last().get\_y1() + direction,

                             beginTxt + str.tostring(slicedVol.sum(), format.volume),

             style = label.style\_none, textcolor = col, size = size.small)

        else

            liveLabel.set\_xy(bar\_index + 5, line.all.last().get\_y1() + direction)

            liveLabel.set\_text(beginTxt + str.tostring(slicedVol.sum(), format.volume))

            liveLabel.set\_color(col)

        cvd = zigZagData.new(cvdCoords = array.new<**chart.point**>())

        normVol = slicedVol.copy()

        if normVol.size() > 1

            for i = normVol.size() - 2 to 0

                normVol.set(i, normVol.get(i + 1) + normVol.get(i))

            lowDelta = normVol.min(), deltaRange = normVol.range(), priceMin = math.min(getY1, getY2, zzd.min)

            absRange = math.abs(priceRange), upper = priceMin + absRange \* .75, lower = priceMin + absRange \* .25

            for i = normVol.size() - 1 to 0

                normed = lower + ((normVol.get(i) - lowDelta) \* (upper - lower)) / deltaRange

                cvd.cvdCoords.push(chart.point.from\_time(zzd.indexArr.get(i), normed))

                livePoly.delete()

                livePoly := polyline.new(cvd.cvdCoords, line\_color = lineCol, xloc = xloc.bar\_time, curved = true, line\_style = sty,

                                                     line\_width = wid

                                                     )

zzd.volArr.drawHistoricalCVD()

zzd.volArr.drawLiveCVD      ()