

//**@version=4**

study("Order Block Finder", overlay = true)

tip1 = "Indicator to help identify instituational Order Blocks (OB). OBs often signal the beginning of a strong move. There is a high probability that OB price levels will be revisited in the future and are interesting levels to place limit orders. Bullish Order block is the last down candle before a sequence of up candles. Bearish Order Block is the last up candle before a sequence of down candles."

tip2 = "!Experimental!\nFind Order Blocks from different timeframes. Channels prices are accurate, but arrow position is not. Most useful when selecting a timeframe higher than the chart."

tip3 = "Required number of subsequent candles in the same direction to identify Order Block"

tip4 = "Measured from from potential OB close to close of first candle in sequence"

dummy = input(true,"Hover over ( ! ) for documentation", tooltip = tip1)

colors = input("LIGHT","Color Scheme", options=["DARK", "LIGHT"])

//res = input("","Order Block Timeframe",input.resolution,tooltip=tip2)

periods = input(7, "Relevant Periods to identify OB",tooltip=tip3) // Required number of subsequent candles in the same direction to identify Order Block

threshold = input(0.0, "Min. Percent move for valid OB", step = 0.1, tooltip=tip4) // Required minimum % move (from potential OB close to last subsequent candle to identify Order Block)

bull\_channels = input(2, "Number of Bullish Channels to show") // Num of channels

bear\_channels = input(2, "Number of Bearish Channels to show") // Num of channels

//Data Curation

res = ""

[copen,chigh,clow,cclose] = security(syminfo.tickerid,res,[open,high,low,close],barmerge.gaps\_on, barmerge.lookahead\_off)

ob\_period = periods + 1 // Identify location of relevant Order Block candle

absmove = ((abs(cclose[ob\_period] - cclose[1]))/cclose[ob\_period]) \* 100 // Calculate absolute percent move from potential OB to last candle of subsequent candles

relmove = absmove >= threshold // Identify "Relevant move" by comparing the absolute move to the threshold

// Color Scheme

bullcolor = colors == "DARK"? color.white : color.green

bearcolor = colors == "DARK"? color.blue : color.red

// Bullish Order Block Identification

bullishOB = cclose[ob\_period] < copen[ob\_period] // Determine potential Bullish OB candle (red candle)

int upcandles = 0

for i = 1 to periods

upcandles := upcandles + (cclose[i] > copen[i]? 1 : 0) // Determine color of subsequent candles (must all be green to identify a valid Bearish OB)

OB\_bull = bullishOB and (upcandles == (periods)) and relmove // Identification logic (red OB candle & subsequent green candles)

OB\_bull\_chigh = OB\_bull? chigh[ob\_period] : na // Determine OB upper limit (Open or High depending on input)

OB\_bull\_clow = OB\_bull? clow[ob\_period] : na // Determine OB clower limit (Low)

OB\_bull\_avg = (OB\_bull\_chigh + OB\_bull\_clow)/2 // Determine OB middle line

// Bearish Order Block Identification

bearishOB = cclose[ob\_period] > copen[ob\_period] // Determine potential Bearish OB candle (green candle)

int downcandles = 0

for i = 1 to periods

downcandles := downcandles + (cclose[i] < copen[i]? 1 : 0) // Determine color of subsequent candles (must all be red to identify a valid Bearish OB)

OB\_bear = bearishOB and (downcandles == (periods)) and relmove // Identification logic (green OB candle & subsequent green candles)

OB\_bear\_chigh = OB\_bear? chigh[ob\_period] : na // Determine OB upper limit (High)

OB\_bear\_clow = OB\_bear? clow[ob\_period] : na // Determine OB clower limit (Open or Low depending on input)

OB\_bear\_avg = (OB\_bear\_clow + OB\_bear\_chigh)/2 // Determine OB middle line

// Plotting

plotshape(OB\_bull, title="Bullish OB", style = shape.triangleup, color = bullcolor, textcolor = bullcolor, size = size.tiny, location = location.belowbar, offset = -ob\_period, text = "Bull") // Bullish OB Indicator

bull1 = plot(OB\_bull\_chigh, title="Bullish OB High", style = plot.style\_linebr, color = bullcolor, offset = -ob\_period, linewidth = 2) // Bullish OB Upper Limit

bull2 = plot(OB\_bull\_clow, title="Bullish OB Low", style = plot.style\_linebr, color = bullcolor, offset = -ob\_period, linewidth = 2) // Bullish OB Lower Limit

fill(bull1, bull2, color=bullcolor, transp = 50, title = "Bullish OB fill") // Fill Bullish OB

plotshape(OB\_bull\_avg, title="Bullish OB Average", style = shape.cross, color = bullcolor, size = size.small, location = location.absolute, offset = -ob\_period) // Bullish OB Average

plotshape(OB\_bear, title="Bearish OB", style = shape.triangledown, color = bearcolor, textcolor = bearcolor, size = size.tiny, location = location.abovebar, offset = -ob\_period, text = "Bear") // Bearish OB Indicator

bear1 = plot(OB\_bear\_clow, title="Bearish OB Low", style = plot.style\_linebr, color = bearcolor, offset = -ob\_period, linewidth = 2) // Bearish OB Lower Limit

bear2 = plot(OB\_bear\_chigh, title="Bearish OB High", style = plot.style\_linebr, color = bearcolor, offset = -ob\_period, linewidth = 2) // Bearish OB Upper Limit

fill(bear1, bear2, color=bearcolor, transp = 50, title = "Bearish OB fill") // Fill Bearish OB

plotshape(OB\_bear\_avg, title="Bearish OB Average", style = shape.cross, color = bearcolor, size = size.small, location = location.absolute, offset = -ob\_period) // Bullish OB Average

var LineBullAvg = array.new\_line()

var LineBullHigh = array.new\_line()

var LineBullLow = array.new\_line()

var LineBearAvg = array.new\_line()

var LineBearHigh = array.new\_line()

var LineBearLow = array.new\_line()

sync = time\_close(res)

if OB\_bull and bull\_channels>0

if array.size(LineBullAvg) == bull\_channels

line.delete(array.get(LineBullAvg,0))

array.remove(LineBullAvg,0)

line.delete(array.get(LineBullHigh,0))

array.remove(LineBullHigh,0)

line.delete(array.get(LineBullLow,0))

array.remove(LineBullLow,0)

array.push(LineBullAvg,line.new(x1 = sync, xloc=xloc.bar\_time, y1 = OB\_bull\_avg, x2 = sync[1], y2 = OB\_bull\_avg, extend = extend.left, color = bullcolor, style = line.style\_solid, width = 1))

array.push(LineBullHigh,line.new(x1 = sync, xloc=xloc.bar\_time,y1 = OB\_bull\_chigh, x2 = sync[1], y2 = OB\_bull\_chigh, extend = extend.left, color = bullcolor, style = line.style\_dashed, width = 1))

array.push(LineBullLow,line.new(x1 = sync, xloc=xloc.bar\_time, y1 = OB\_bull\_clow, x2 = sync[1], y2 = OB\_bull\_clow, extend = extend.left, color = bullcolor, style = line.style\_dashed, width = 1))

if OB\_bear and bear\_channels>0

if array.size(LineBearAvg) == bear\_channels

line.delete(array.get(LineBearAvg,0))

array.remove(LineBearAvg,0)

line.delete(array.get(LineBearHigh,0))

array.remove(LineBearHigh,0)

line.delete(array.get(LineBearLow,0))

array.remove(LineBearLow,0)

array.push(LineBearAvg,line.new(x1 = sync, xloc=xloc.bar\_time, y1 = OB\_bear\_avg, x2 = sync[1], y2 = OB\_bear\_avg, extend = extend.left, color = bearcolor, style = line.style\_solid, width = 1))

array.push(LineBearHigh,line.new(x1 = sync, xloc=xloc.bar\_time, y1 = OB\_bear\_chigh, x2 = sync[1], y2 = OB\_bear\_chigh, extend = extend.left, color = bearcolor, style = line.style\_dashed, width = 1))

array.push(LineBearLow,line.new(x1 = sync, xloc=xloc.bar\_time, y1 = OB\_bear\_clow, x2 = sync[1], y2 = OB\_bear\_clow, extend = extend.left, color = bearcolor, style = line.style\_dashed, width = 1))

// Alerts for Order Blocks Detection

alertcondition(OB\_bull, title='New Bullish OB detected', message='New Bullish OB detected - This is NOT a BUY signal!')

alertcondition(OB\_bear, title='New Bearish OB detected', message='New Bearish OB detected - This is NOT a SELL signal!')