

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

//@version=6

indicator('MFB - Footprints', overlay = true, max_boxes_count = 500, max_labels_count =
500, max_lines_count = 500)

// --- Colors & Constants ---

COLOR_BULL = #089981

COLOR_BEAR = #f23645

COLOR_ENTRY = #5b9cf6

COLOR_DASH = color.new(color.black, 40)

// --- Inputs ---

grpHTF = '🌐 HTF Filter Settings'

use_15m = input.bool(true, 'Include 15m FVG', group = grpHTF)

use_1h = input.bool(true, 'Include 1h FVG', group = grpHTF)

use_4h = input.bool(true, 'Include 4h FVG', group = grpHTF)

grpLevels = '🟣 Key Level Integration (TP Snap)'

use_snap = input.bool(true, 'Snap TP to Key Levels (PDH/PDL/PWH/PWL)', group =
grpLevels)

grpStyle = '🎨 Style & Cleanup'

lookback_hrs = input.int(48, 'Show Trades from Last (Hours)', minval = 1, group = grpStyle)

show_fvg_boxes = input.bool(true, 'Show Active 5m FVG Boxes', group = grpStyle)

fvg_transp = input.int(90, 'FVG Box Transparency', minval = 0, maxval = 100, group =
grpStyle)

```
line_len = input.int(15, 'Management Line Length', minval = 5, group = grpStyle)
```

```
// --- HTF Data & Levels ---
```

```
pdh = request.security(syminfo.tickerid, 'D', high[1], lookahead = barmerge.lookahead_on)
```

```
pdl = request.security(syminfo.tickerid, 'D', low[1], lookahead = barmerge.lookahead_on)
```

```
pwh = request.security(syminfo.tickerid, 'W', high[1], lookahead = barmerge.lookahead_on)
```

```
pwl = request.security(syminfo.tickerid, 'W', low[1], lookahead = barmerge.lookahead_on)
```

```
f_get_hft_context() =>
```

```
var float fvg_t = na
```

```
var float fvg_b = na
```

```
var int is_bull_int = 0
```

```
var int state = 0
```

```
var float rr1_target = na
```

```
var float sl_level = na
```

```
bull_fvg = low > high[2]
```

```
bear_fvg = high < low[2]
```

```
if bull_fvg
```

```
    fvg_t := low
```

```
    fvg_b := high[2]
```

```
    is_bull_int := 1
```

```
    state := 1
```

```
else if bear_fvg
```

```
    fvg_t := low[2]
```

```
    fvg_b := high
```

```
    is_bull_int := -1
```

```

state := 1
if state == 1
  if (is_bull_int == 1 and low <= fvg_t) or (is_bull_int == -1 and high >= fvg_b)
    state := 2
if state == 2
  if (is_bull_int == 1 and close > fvg_t) or (is_bull_int == -1 and close < fvg_b)
    state := 3
    sl_level := is_bull_int == 1 ? fvg_b : fvg_t
    risk = math.abs(close - sl_level)
    rr1_target := is_bull_int == 1 ? close + risk : close - risk
  in_fvg = close <= fvg_t and close >= fvg_b
  in_rr1 = is_bull_int == 1 ? (close >= sl_level and close <= rr1_target) : (close <= sl_level and
close >= rr1_target)
  context_active = (in_fvg or in_rr1) and not na(fvg_t)
  [context_active, is_bull_int]

[ctx15, dir15] = request.security(syminfo.tickerid, '15', f_get_htf_context())
[ctx1h, dir1h] = request.security(syminfo.tickerid, '60', f_get_htf_context())
[ctx4h, dir4h] = request.security(syminfo.tickerid, '240', f_get_htf_context())

bool htf_bullish = (use_15m and ctx15 and dir15 == 1) or (use_1h and ctx1h and dir1h == 1)
or (use_4h and ctx4h and dir4h == 1)

bool htf_bearish = (use_15m and ctx15 and dir15 == -1) or (use_1h and ctx1h and dir1h == -
1) or (use_4h and ctx4h and dir4h == -1)

// --- Arrays & Variables ---
var array<box> fvg_boxes = array.new_box()

```

```
var array<float> fvg_top    = array.new_float()
var array<float> fvg_bottom  = array.new_float()
var array<bool> fvg_is_bull  = array.new_bool()
var array<int> fvg_state    = array.new_int()
var array<int> fvg_created_id = array.new_int()
var array<box> trade_boxes_tp = array.new_box()
var array<box> trade_boxes_sl = array.new_box()
var array<line> trade_lines_ep = array.new_line()
var array<line> trade_lines_sl = array.new_line()
var array<line> trade_lines_tp = array.new_line()
var array<int> trade_times   = array.new_int()
```

```
var string last_anchor_tf = "None"
var bool is_long_alert = false
var bool is_short_alert = false
var bool is_coach_signal = false
```

```
// Reset alerts each bar
```

```
is_long_alert := false
is_short_alert := false
is_coach_signal := false
```

```
if timeframe.isintraday and timeframe.multiplier == 5
    if low > high[2]
        array.push(fvg_top, low)
        array.push(fvg_bottom, high[2])
```

```

    array.push(fvg_is_bull, true)

    array.push(fvg_state, 0)

    array.push(fvg_created_id, bar_index)

    if show_fvg_boxes

        array.push(fvg_boxes, box.new(bar_index[2], low, bar_index, high[2], bgcolor =
color.new(COLOR_BULL, fvg_transp), border_color = color.new(COLOR_BULL,
fvg_transp)))

    else

        array.push(fvg_boxes, box(na))

    if high < low[2]

        array.push(fvg_top, low[2])

        array.push(fvg_bottom, high)

        array.push(fvg_is_bull, false)

        array.push(fvg_state, 0)

        array.push(fvg_created_id, bar_index)

        if show_fvg_boxes

            array.push(fvg_boxes, box.new(bar_index[2], low[2], bar_index, high, bgcolor =
color.new(COLOR_BEAR, fvg_transp), border_color = color.new(COLOR_BEAR,
fvg_transp)))

        else

            array.push(fvg_boxes, box(na))

    int i = array.size(fvg_top) - 1

    while i >= 0

        float ft = array.get(fvg_top, i)

        float fb = array.get(fvg_bottom, i)

        bool fl = array.get(fvg_is_bull, i)

```

```

int fs = array.get(fvg_state, i)

int fc = array.get(fvg_created_id, i)

box b = array.get(fvg_boxes, i)

if not na(b)

    box.set_right(b, bar_index)

if (fl and low < fb) or (not fl and high > ft)

    if not na(b)

        box.delete(b)

    array.remove(fvg_boxes, i)

    array.remove(fvg_top, i)

    array.remove(fvg_bottom, i)

    array.remove(fvg_is_bull, i)

    array.remove(fvg_state, i)

    array.remove(fvg_created_id, i)

    i := i - 1

    continue

if fs == 0 and bar_index > fc and ((fl and low <= ft) or (not fl and high >= fb))

    array.set(fvg_state, i, 1)

if fs == 1 and bar_index > fc

    if (fl ? close > ft : close < fb) and (fl ? htf_bullish : htf_bearish)

        float entry_p = close

        float stop_p = fl ? fb : ft

        float risk_v = math.abs(entry_p - stop_p)

        float target_p = fl ? entry_p + (risk_v * 2) : entry_p - (risk_v * 2)

        if use_snap

            if fl

```

```

    if pdh > entry_p and target_p > pdh
        target_p := pdh
    if pwh > entry_p and target_p > pwh
        target_p := pwh
    else
        if pdl < entry_p and target_p < pdl
            target_p := pdl
        if pwl < entry_p and target_p < pwl
            target_p := pwl

    label.new(bar_index, fl ? low : high, fl ? "L" : "S", style = fl ? label.style_label_up :
label.style_label_down, color = fl ? COLOR_BULL : COLOR_BEAR, textcolor = color.white,
size = size.small)

    array.push(trade_boxes_tp, box.new(bar_index, fl ? target_p : entry_p, bar_index +
line_len, fl ? entry_p : target_p, bgcolor = color.new(COLOR_BULL, 85), border_color =
color.new(COLOR_BULL, 100)))

    array.push(trade_boxes_sl, box.new(bar_index, fl ? entry_p : stop_p, bar_index +
line_len, fl ? stop_p : entry_p, bgcolor = color.new(COLOR_BEAR, 85), border_color =
color.new(COLOR_BEAR, 100)))

    array.push(trade_lines_ep, line.new(bar_index, entry_p, bar_index + line_len,
entry_p, color = COLOR_ENTRY, style = line.style_dotted, width = 2))

    array.push(trade_lines_sl, line.new(bar_index, stop_p, bar_index + line_len, stop_p,
color = COLOR_BEAR, style = line.style_dotted, width = 2))

    array.push(trade_lines_tp, line.new(bar_index, target_p, bar_index + line_len,
target_p, color = COLOR_BULL, style = line.style_dotted, width = 2))

    array.push(trade_times, time)

    last_anchor_tf := ctx4h ? "4h" : ctx1h ? "1h" : ctx15 ? "15m" : "None"

    if fl
        is_long_alert := true
    else

```

```

        is_short_alert := true

    if (use_1h and ctx1h) or (use_15m and ctx15)

        is_coach_signal := true

    if not na(b)

        box.delete(b)

    array.remove(fvg_boxes, i)

    array.remove(fvg_top, i)

    array.remove(fvg_bottom, i)

    array.remove(fvg_is_bull, i)

    array.remove(fvg_state, i)

    array.remove(fvg_created_id, i)

    i := i - 1

int k = array.size(trade_times) - 1
while k >= 0

    if (time - array.get(trade_times, k)) > (lookback_hrs * 3600000)

        box.delete(array.remove(trade_boxes_tp, k))

        box.delete(array.remove(trade_boxes_sl, k))

        line.delete(array.remove(trade_lines_ep, k))

        line.delete(array.remove(trade_lines_sl, k))

        line.delete(array.remove(trade_lines_tp, k))

        array.remove(trade_times, k)

    k := k - 1

// --- Dashboard ---

```



```

var table board = table.new(position.bottom_right, 2, 2, bgcolor = COLOR_DASH,
border_width = 1, border_color = color.gray)

if barstate.islast

    table.cell(board, 0, 0, "Within HTF Footprint:", text_color = color.white, text_size =
size.large)

    table.cell(board, 1, 0, (htf_bullish or htf_bearish ? "Yes" : "No"), text_color = (htf_bullish or
htf_bearish ? COLOR_BULL : color.gray), text_size = size.large)

    table.cell(board, 0, 1, "Last 5m Signal Anchor:", text_color = color.white, text_size =
size.large)

    table.cell(board, 1, 1, last_anchor_tf, text_color = COLOR_ENTRY, text_size = size.large)


// --- Alert Triggers ---

alertcondition(is_coach_signal, "Coach's Favorite Signal", "Setup confirmed")
alertcondition(is_long_alert, "LONG setup", "LONG confirmed")
alertcondition(is_short_alert, "SHORT setup", "SHORT confirmed")
alertcondition(is_long_alert or is_short_alert, "Any setup", "Setup confirmed")

```