

Алгоритм индикатора MasterForex-V для MetaTrader 5

Обзор индикатора и принципов MasterForex-V

MasterForex-V индикатор – это мульти-таймфреймовый инструмент анализа, который одновременно обрабатывает данные с нескольких периодов (H1, M15, M5) и отображает на одном графике ключевые уровни и торговые сигналы. Такой подход решает проблему постоянного переключения таймфреймов для оценки общей картины рынка ¹. Индикатор отображает уровни **MF-pivot** (значимые экстремумы) на каждом ТФ, определяет направление тренда (Up/Down/Flat) и генерирует графические сигналы: стрелки входа, значки выхода и метки разворота тренда. Ниже подробно рассмотрено, **что** отображается на графике и **как** рассчитывается каждый элемент, а также приведен план реализации этого алгоритма в MQL5.

Мульти-таймфреймовый ZigZag и расчет уровней MF-pivot

1. Получение экстремумов (ZigZag) на H1, M15, M5: На каждом из трех таймфреймов сначала определяются ключевые локальные максимумы и минимумы с помощью индикатора ZigZag. ZigZag соединяет значимые вершины и впадины цены, отфильтровывая мелкие колебания. В MQL5 можно использовать встроенный индикатор ZigZag через `iCustom` либо реализовать свой алгоритм. Например, инициализация ZigZag для разных ТФ (с параметрами Depth=12, Deviation=5, Backstep=3) и извлечение экстремумов может выглядеть так:

```
// Инициализация ZigZag для H1, M15, M5
zigzagH1 = iCustom(_Symbol, PERIOD_H1, "ZigZag", 12, 5, 3);
zigzagM15 = iCustom(_Symbol, PERIOD_M15, "ZigZag", 12, 5, 3);
zigzagM5 = iCustom(_Symbol, PERIOD_M5, "ZigZag", 12, 5, 3);

// Копируем буфер ZigZag и извлекаем точки экстремумов на M5:
double zigzagBufferM5[];
CopyBuffer(zigzagM5, 0, 0, Bars(_Symbol, PERIOD_M5), zigzagBufferM5);

// Фильтруем EMPTY_VALUE – ZigZag ставит цену только на барах-экстремумах:
double pivotsM5[];
for(int i=0; i < ArraySize(zigzagBufferM5); i++) {
    if(zigzagBufferM5[i] != EMPTY_VALUE) {
        pivotsM5.push_back(zigzagBufferM5[i]);
    }
}
// Теперь массив pivotsM5 содержит цены всех экстремумов (последний элемент – последний экстремум).
```

Зигзаговый индикатор заполняет свой буфер специальным значением `EMPTY_VALUE` на всех барах, **кроме** тех, где обнаружен локальный максимум или минимум. Поэтому после `CopyBuffer`

необходимо отфильтровать пустые значения – оставшиеся и будут ценами экстремумов ². Аналогично получаем массивы `pivotsH1` и `pivotsM15` для других таймфреймов. Последние элементы таких массивов представляют наиболее свежие (ближайшие к текущей цене) экстремумы на каждом ТФ.

2. Выбор значимого экстремума (уровень MF-pivot): На базе полученных экстремумов вычисляется уровень **MF-pivot** – **первая значимая точка коррекции, от которой начался текущий импульс**. Проще говоря, MF-pivot – это уровень предыдущего важного экстремума, который «защищает» текущий тренд. Этот уровень служит границей между фазой импульса и потенциальным разворотом/флэтом. Алгоритм определения MF-pivot на каждом таймфрейме такой:

- Если последним сформированным экстремумом является локальный **минимум** (цена сделала впадину и развернулась вверх), то этот минимум и есть MF-pivot для восходящего тренда на данном ТФ (уровень поддержки начала импульса).
- Если последний экстремум – локальный **максимум** (цена достигла вершины и развернулась вниз), то его уровень выступает как MF-pivot для нисходящего тренда (уровень сопротивления начала импульса).
- Если на таймфрейме нет четко выраженного импульса (движение идет в боковом диапазоне, волны маловыражены), то определить значимый экстремум затруднительно – в таких случаях тренд считается неопределенным (Flat), а pivot временно может не использоваться либо не обновляться.

Алгоритмически это можно реализовать, сравнив последние два экстремума из массива `pivotsTF`. Пусть `lastPivot` – последний экстремум (ближайший к текущей цене), а `prevPivot` – предыдущий. Тогда:

- Если `lastPivot` **ниже** `prevPivot` по цене, значит последняя образованная ZigZag-точка – это локальный минимум. Цена сделала впадину и пошла вверх, следовательно `lastPivot` становится уровнем **MF-pivot** для восходящего тренда на данном ТФ (начало текущего импульса вверх).
- Если `lastPivot` **выше** `prevPivot`, то последний экстремум – локальный максимум. Цена достигла вершины и развернулась вниз – этот уровень будет **MF-pivot** для нисходящего тренда (начало текущего импульса вниз).

Таким образом определяем **значимый уровень pivot** на каждом таймфрейме. (Примечание: если ZigZag еще не отрисовал новую точку на самом последнем баре, стоит брать предыдущий завершённый экстремум как актуальный pivot, чтобы избежать неопределенности из-за перерисовки ZigZag.)

Определение направления тренда на каждом ТФ

После вычисления MF-pivot на каждом таймфрейме определяем актуальный тренд следующим образом:

- **Uptrend (восходящий):** если текущая цена **выше** уровня MF-pivot и последняя завершённая волна ZigZag была восходящей (последний экстремум – минимум). Иными словами, пока цена держится выше своего ключевого предыдущего минимума, структура максимумов/минимумов повышается – тренд считается восходящим ³ ⁴.

- **Downtrend (нисходящий):** если цена **ниже** уровня MF-pivot **и** последняя волна была нисходящей (последний экстремум – максимум). Пока цена не поднялась выше последнего значимого максимума, тренд на этом ТФ сохраняется нисходящим.
- **Flat (боковой рынок):** если цена блуждает **около** уровня pivot, периодически пробивая его, но без развития нового импульса в какую-либо сторону. Также флэт фиксируется, если амплитуда колебаний мала – например, диапазон цен за последние N баров очень узкий или средний True Range ниже заданного порога (низкая волатильность). В этих случаях направление тренда не выражено, сигналы индикатора лучше игнорировать.

На практике, в коде после расчета pivot можно просто сравнить текущую цену с уровнем:

```
if(latestPrice > pivotH1) trendH1 = "Up";
else if(latestPrice < pivotH1) trendH1 = "Down";
else trendH1 = "Flat";
// (Аналогично для M15, M5)
```

(Замечание: Более точно учитывать тренд позволяет также анализ последней волны ZigZag, как описано выше, но зачастую сравнения с pivot уже достаточно.) Если на старшем ТФ тренд определяется как Flat, индикатор либо не генерирует стрелки вовсе, либо может давать только предупредительные слабые сигналы.

Элементы индикатора на графике

Индикатор визуализирует на графике сразу несколько элементов со всех трех таймфреймов, согласно описанию:

Горизонтальные линии уровней MF-pivot (H1, M15, M5)

На графике проводятся горизонтальные линии, показывающие текущий уровень MF-pivot на каждом таймфрейме. Для наглядности используется разный цвет: например, синий – уровень pivot с H1, зеленый – с M15, оранжевый – с M5. Эти линии автоматически перемещаются по мере появления новых экстремумов и сдвига ключевых точек (то есть при обновлении значений pivot):

- **Реализация:** создать три объекта типа `OBJ_HLINE` (горизонтальная линия) – по одному для каждого ТФ – либо использовать индикаторные буферы с постоянным значением на уровне pivot. Проще управлять объектами: при каждом пересчете индикатора достаточно обновить позицию линий через функцию `ObjectSetDouble` (или `ObjectMove`). Имя и цвет каждой линии фиксируются (например, `"PivotH1"`, `"PivotM15"`, `"PivotM5"`). При обновлении вызывается:

```
ObjectSetDouble(0, "PivotH1", OBJPROP_PRICE, pivotH1);
```

что переместит линию на новый уровень ⁵ (здесь `chart_id=0` означает текущий график). Аналогично обновляются линии для M15 и M5.

Волновая разметка ZigZag (рабочий ТФ, обычно M5)

Для наглядного отображения структуры волн на графике рисуется ломаная линия ZigZag, соединяющая локальные максимумы и минимумы цены. Как правило, разметку имеет смысл

делать на младшем таймфрейме (M5), где видны все колебания внутри часа. Это позволяет трейдеру визуально сопоставлять стрелки сигналов с волновой картиной рынка (видно, где были последние вершины/впадины и как формируется импульс).

- **Реализация:** можно отобразить ZigZag с помощью индикаторных буферов типа `DRAW_SECTION` – достаточно проставить значение цены на тех барах, где есть экстремум, а на остальных барах заполнить `EMPTY_VALUE`. MetaTrader автоматически соединит такие точки линиями. В нашем случае, получив массив экстремумов `pivotsM5` (ценовые значения на барах-экстремумах), можно скопировать его в специальный буфер индикатора для отрисовки. Альтернативный способ – создавать графические объекты (например, отдельные линии `OBJ_TREND`) между последовательными экстремумами, но это сложнее в управлении, поэтому использование буфера предпочтительнее и проще.

Стрелки входа (Buy/Sell)

Индикатор отмечает на графике стрелками места потенциального входа в сделки на покупку или продажу. Стрелка **вверх** (Buy) появляется **под баром**, когда сформирован сигнал на покупку; стрелка **вниз** (Sell) – **над баром** при сигнале на продажу. Цвет и прозрачность стрелки отражают силу сигнала:

- **Яркая стрелка насыщенного цвета** – основной *подтвержденный* сигнал. Возникает, когда тренды на всех трех таймфреймах совпадают (полное согласие направления Up или Down) и младший ТФ (M5) подтвердил импульс пробоем локального экстремума в сторону тренда. Например, если H1, M15, M5 все показывают восходящий тренд, и на M5 завершилась коррекция (небольшой откат) и цена пробила предыдущий локальный максимум вверх – рисуется яркая стрелка **Buy** ⁶ ⁷ под текущей свечой. Это сигнал на покупку по тренду.
- **Полупрозрачная (бледная) стрелка** – ранний, агрессивный сигнал. Появляется, когда **только младший** ТФ развернулся, а старшие еще не подтвердили новый тренд. Это возможно, например, если H1 уже был восходящий, но M15 оставался нисходящим – и вот M5 первым показал разворот вверх (цена на M5 пробила свой локальный уровень сопротивления). В таком случае индикатор отобразит бледную стрелку вверх под баром: сигнал о возможном начале разворота, который пока подтвержден лишь на M5 ⁸ ⁹. Аналогично для продаж: скажем, H1 Downtrend, M15 еще Up, но M5 развернулся вниз – появится полупрозрачная стрелка **Sell** над соответствующим баром.
- **Фильтрация флэта:** если на каком-либо старшем таймфрейме тренд определен как Flat, сигналы либо не подаются вовсе, либо могут быть только ранними (с пониженной достоверностью). Это предохраняет от ложных входов в вялом, боковом рынке.

Алгоритм генерации стрелок (логика): пусть `trendH1`, `trendM15`, `trendM5` – вычисленные на текущем тике направления тренда для H1, M15, M5 соответственно. Индикатор должен отследить момент, когда M5 завершает локальную коррекцию и возобновляет движение по тренду. Практически, сигнал фиксируется по **смене направления** `trendM5` с противоположного на совпадающее со старшими. Например:

- Если на предыдущем шаге `trendM5` был не "Up" (был "Down" или "Flat"), а сейчас `trendM5 = "Up"` (произошел разворот вверх на M5):
- **Если при этом** `trendH1 == "Up"` **и** `trendM15 == "Up"`, то все три ТФ в унисон вверх – появляется **яркая зеленая стрелка Buy** под текущей свечой M5 как основной сигнал.
- **Если** `trendH1 == "Up"`, а `trendM15` еще **не** "Up" (т.е. M15 в состоянии Down или Flat), то младший ТФ подал первый сигнал вверх при общем восходящем настрое старшего ТФ –

рисуются **полупрозрачная стрелка вверх** (ранний сигнал, подтвержденный H1 и M5, но не M15).

- Аналогично: если M5 развернулся **вниз** (`trendM5` сменился на "Down");
- Если `trendH1 == "Down"` и `trendM15 == "Down"`, рисуем **яркую красную стрелку Sell** над текущим баром.
- Если `trendH1 == "Down"`, а `trendM15` еще не перешел в Down (остается Up или Flat), рисуется **бледная стрелка Sell** (ранний сигнал вниз, подтвержден только H1 и M5).
- Если же самый старший таймфрейм H1 **не совпадает** по тренду с сигналом M5 (например, H1 Down, а M5 подал Up), **стрелка не рисуется вовсе** – такой сигнал противоречит глобальному тренду и игнорируется. Индикатор требует хотя бы на высшем уровне согласованности направления для входа.

После появления стрелки входа рекомендуется выставить стоп-лосс за ближайшим pivot на M5 (этот уровень индикатор знает и может, например, вывести во всплывающей подсказке). Такая тактика логична: если цена развернется обратно и пробьет этот уровень, значит сигнал стал невалидным. Индикатор может помогать в этом – например, отображать уровень стопа или упоминать его значение при формировании сигнала.

- **Реализация стрелок:** для отрисовки стрелок можно воспользоваться графическими объектами. В MQL5 есть предустановленный тип `OBJ_ARROW` (стрелка) или можно использовать `OBJ_LABEL` с символом Wingdings (набор специальных символов). При создании объекта важно задать корректные координаты времени и цены: стрелку **Buy** размещаем чуть ниже минимума соответствующего бара, стрелку **Sell** – чуть выше максимума бара. Каждой стрелке даем уникальное имя (например, включающее время или индекс бара). Цвет: для **ярких сигналов** устанавливаем сплошной цвет (например, ярко-зеленый для Buy, красный для Sell), для **ранних** – тот же цвет, но с прозрачностью ~50% (через альфа-канал) или более светлым оттенком. Например, в MQL5 можно задать цвет с альфа-каналом: `ColorToARGB clrLime, 128` – получим примерно 50% прозрачности у зеленого цвета. Пример кода для стрелки Buy:

```
string name = "BuyArrow_"+(string)TimeCurrent();
ObjectCreate(0, name, OBJ_ARROW, 0, TimeCurrent(), Low[0] - 10*Point); // на
текущем баре, чуть ниже Low
ObjectSetInteger(0, name, OBJPROP_ARROWCODE, 233); // код символа стрелки
вверх (Wingdings 233)
ObjectSetInteger(0, name, OBJPROP_COLOR, ColorToARGB(clrLime, 128)); //
полупрозрачный зеленый
ObjectSetInteger(0, name, OBJPROP_WIDTH, 1);
```

В этом примере стрелка разместится под текущей свечой. Для Sell-стрелки аналогично, но координата по цене `High[0] + небольшой отступ`. Параметр `Point` используется, чтобы отступить несколько пунктов и стрелка не налезала на сам бар.

Значок выхода (рекомендация закрыть позицию)

Когда после входа в сделку появляются признаки, что тенденция исчерпала себя, индикатор помечает рекомендуемый момент **выхода** специальным значком – например, полупрозрачным крестиком "X" или иконкой двери/выхода. Значок ставится над либо под тем баром, на котором выполнены условия для закрытия. Если трейдер все еще в позиции после сигнала входа,

появление такого крестика подсказывает: пора фиксировать прибыль или полностью выходить. Основные критерии для сигнала выхода могут быть следующие:

- **Пробой уровня pivot в обратную сторону.** Цена пересекла ключевой уровень, защищавший предыдущий импульс, что означает слом тенденции. Например, при открытой Buy-сделке (тренд был Up) цена резко развернулась вниз и пробила **вниз** актуальный pivot на M5 (наш уровень стоп-лосса) или даже pivot M15. Это сигнал, что восходящий импульс исчерпан. Индикатор отметит на этом баре крестик **✕** (иконка выхода) – показывая, что тренд сломлен и позицию лучше закрыть.
- **Достижение целевого ориентира.** Например, достижение важных уровней поддержки/сопротивления: дневных уровней S1/R1 (пивотных уровней), либо рассчитанных целей по Фибоначчи, либо других известных целей профита. В описании алгоритма упоминалось, что при достижении цели индикатор тоже может сигнализировать на выход. (В базовой реализации можно для упрощения опустить эту часть или задать фиксированный уровень профита, чтобы не усложнять код.)
- **Сигналы ослабления тренда.** Например, дивергенция на осцилляторах (АО, RSI и др.), появление разворотных свечных паттернов (доджи, пин-бар) или резкий всплеск объема против позиции. Эти фильтры более сложные и могут быть добавлены при необходимости для подтверждения выхода, но в простейшем индикаторе можно обойтись без них.

Алгоритм выхода (упрощенно): после каждой стрелки входа индикатор может запомнить направление открытой позиции (`openPositionSide = Buy/Sell`). Далее он на каждом шаге отслеживает условия:

- Для **лонга (Buy)**: если образуется новый pivot на M5 и затем цена пробивает его **вниз** (т.е. опускается ниже последнего значимого минимума), **либо** тренд на среднем ТФ (M15) развернулся в Down – пометить бар пробоя значком выхода. (Фактически, это срабатывание предполагаемого стоп-лосса или начало нисходящего импульса старшего порядка).
- Для **шорта (Sell)**: если появился новый pivot M5 и затем цена пробивает его **вверх** (выше последнего максимума), **либо** тренд M15 развернулся в Up – вывести значок выхода.

Также можно задать дополнительный критерий: фиксированный профит-уровень. Например, двойной размер риска (Risk:Reward = 1:2) или близлежащий уровень R1/S1. Если цена его достигла, индикатор ставит значок выхода на этом баре (трейдер может решить закрыть позицию полностью или частично на таком уровне).

- **Реализация значка выхода:** используем, например, объект `OBJ_LABEL` с кодом символа Wingdings для крестика (например, код 251 – символ “✕”). Цвет делаем нейтральным (серый или светлый), иконку полупрозрачной, чтобы не загромождала график. Размещаем над свечой для сигнала продажи (чуть выше High бара), или под свечой для сигнала покупки (чуть ниже Low). Каждому значку даем уникальное имя, как и стрелкам. Пример для выхода (крестик) при продаже:

```
string name = "Exit_"+(string)TimeCurrent();
ObjectCreate(0, name, OBJ_LABEL, 0, TimeCurrent(), High[0] + 10*Point);
ObjectSetInteger(0, name, OBJPROP_FONT, FONT_WINGDINGS);
ObjectSetInteger(0, name, OBJPROP_FONTSIZE, 8);
ObjectSetInteger(0, name, OBJPROP_COLOR, ColorToARGB clrSilver, 150)); //
```

полупрозрачный серый

```
ObjectSetString(0, name, OBJPROP_TEXT, "x");
```

Знак разворота тренда

Особый символ на графике, сигнализирующий о **смене глобального тренда**. Он появляется, когда старший таймфрейм подтверждает разворот. В терминах алгоритма – пробит pivot старшего ТФ и сформирован новый импульс на среднем ТФ. Например, был долгий нисходящий тренд на H1, защищенный своим pivot. Цена развернулась вверх и пробила pivot H1, указывая на смену тенденции, плюс на M15 уже прорисовалась новая волна роста (цена закрепилась выше своего pivot). Тогда индикатор ставит, скажем, изогнутую стрелку вверх в районе этого события – мол, тренд развернулся вверх. Это важно, потому что после подтвержденного разворота меняется тактика торговли (следует искать покупки вместо продаж, или наоборот).

- **Реализация:** индикатор при каждом тике/баре проверяет, не изменилось ли состояние `trendH1` по сравнению с предыдущим значением. Если да (например, было "Down", стало "Up") и при этом `trendM15` совпадает с новым направлением (то есть M15 тоже уже перешел в "Up"), срабатывают условия **разворота вверх**. Рисуем на графике значок разворота – можно выбрать специальный символ (например, Wingdings-стрелку с изгибом, либо сочетание дуги и стрелки). Цвет – зеленый для разворота вверх. Если H1 сменился с "Up" на "Down" и M15 также "Down", рисуем значок разворота вниз (цвет красный).

Такой значок рисуется **единожды** в момент смены тенденции. Чтобы не дублировать его на каждом баре, можно завести флаг состояния или хранить, когда последний раз рисовали разворот, и больше не рисовать тот же самый разворот. Визуально значок можно поместить чуть выше/ниже того экстремума, где произошел пробой pivot, либо над/под соответствующей свечой-пробой. Например, красную загнутую стрелку вниз – над баром, пробившим восходящий pivot, обозначая перелом тренда вниз. Дополнительно можно сопровождать событие разворота уведомлением: всплывающее окно, звуковой сигнал, либо push-уведомление (`Alert()`, `SendNotification()` и т.д.), чтобы трейдер точно не пропустил смену тренда.

Панель статуса трендов

В уголке основного окна графика выводится небольшая текстовая панель, показывающая текущий статус тренда на каждом из трех таймфреймов. Например: **"H1: Up | M15: Up | M5: Flat"**. Эта панель постоянно обновляется, давая быстрое представление о согласованности движения на разных уровнях.

Для удобства можно цветом подсвечивать определенные ситуации: если все три ТФ показывают Up – текст панели сделать зеленым, если все Down – красным; если смешанные (нет единства) – желтым или белым. В панель при желании можно добавлять и дополнительную информацию (например, величину последнего импульса в пунктах, текущее значение ATR для оценки волатильности и т.д.), но в минимальном варианте достаточно отображать только направления трендов.

- **Реализация:** проще всего использовать объект `OBJ_LABEL`, привязанный к окну графика. Создаем его в `OnInit()` с нужными параметрами расположения, например в правом верхнем углу:

```

ObjectCreate(0, "TrendStatus", OBJ_LABEL, 0, 0, 0);
ObjectSetInteger(0, "TrendStatus", OBJPROP_CORNER, CORNER_RIGHT_UPPER);
ObjectSetInteger(0, "TrendStatus", OBJPROP_XDISTANCE, 10);
ObjectSetInteger(0, "TrendStatus", OBJPROP_YDISTANCE, 10);
ObjectSetInteger(0, "TrendStatus", OBJPROP_COLOR, clrWhite);
ObjectSetInteger(0, "TrendStatus", OBJPROP_FONT_SIZE, 10);

```

В `OnCalculate` (или `OnTick`) обновляем текст надписи согласно текущим направлениям:

```

string status = "H1:" + trendH1 + " | M15:" + trendM15 + " | M5:" + trendM5;
ObjectSetString(0, "TrendStatus", OBJPROP_TEXT, status);

```

Таким образом при каждом обновлении графика панель будет показывать актуальные направления тренда. (Для изменения цвета текста в зависимости от ситуации можно вызывать `ObjectSetInteger(OBJPROP_COLOR, ...)` по условию сочетания трендов.)

Логика реализации алгоритма (пошагово)

Ниже приведен обобщенный псевдокод для индикатора в MQL5, объединяющий все описанные этапы логики. Этот алгоритм рассчитан на работу индикатора, запущенного, например, на M5-графике базового инструмента (это оптимально, так как M5 – младший рабочий ТФ, и индикатор сможет получить доступ ко всем необходимым барам H1 и M15 через функции `iCustom/CopyBuffer`). В коде происходит последовательное получение данных ZigZag с разных периодов, вычисление pivot и трендов, затем обновление графических объектов и генерация сигналов:

```

// --- OnInit() ---
// Инициализация индикатора: задание параметров, цветов, буферов и пр.

// Получаем хендлы ZigZag для нужных ТФ (если используем встроенный индикатор ZigZag):
zigzagH1 = iCustom(_Symbol, PERIOD_H1, "ZigZag", Depth, Deviation, Backstep);
zigzagM15 = iCustom(_Symbol, PERIOD_M15, "ZigZag", Depth, Deviation, Backstep);
zigzagM5 = iCustom(_Symbol, PERIOD_M5, "ZigZag", Depth, Deviation, Backstep);

// Создаем объекты для линий pivot и метки статуса:
CreateHLine("PivotH1", clrBlue);
CreateHLine("PivotM15", clrGreen);
CreateHLine("PivotM5", clrOrange);
CreateLabel("TrendStatus");

// --- OnCalculate(...) ---
static string trendH1, trendM15, trendM5;           // текущие тренды
static string prevTrendH1, prevTrendM15, prevTrendM5; // тренды на
// предыдущем расчете
static string openPositionSide = "None";           // фиктивное

```



```

состояние открытой позиции ("Buy","Sell" или None)

// 1. Обновление данных ZigZag на каждом таймфрейме:
UpdateZigZag(zigzagH1, pivotsH1); // CopyBuffer + фильтрация EMPTY_VALUE
для H1
UpdateZigZag(zigzagM15, pivotsM15);
UpdateZigZag(zigzagM5, pivotsM5);

// 2. Вычисление уровней MF-pivot и направлений тренда:
double pivotH1 = CalculatePivot(pivotsH1); // последний значимый
экстремум (low или high)
double pivotM15 = CalculatePivot(pivotsM15);
double pivotM5 = CalculatePivot(pivotsM5);

// Сохраняем предыдущее состояние трендов:
prevTrendH1 = trendH1;
prevTrendM15 = trendM15;
prevTrendM5 = trendM5;

// Определяем текущие тренды (Up/Down/Flat) на каждом ТФ:
trendH1 = DetermineTrend(pivotH1,
Close_price_H1); // (можно использовать последнюю цену H1 или текущую цену)
trendM15 = DetermineTrend(pivotM15, Close_price_M15);
trendM5 = DetermineTrend(pivotM5, Close_price_M5);

// 3. Обновление отображаемых линий pivot на графике:
ObjectSetDouble(0, "PivotH1", OBJPROP_PRICE, pivotH1);
ObjectSetDouble(0, "PivotM15", OBJPROP_PRICE, pivotM15);
ObjectSetDouble(0, "PivotM5", OBJPROP_PRICE, pivotM5);

// 4. Обновление линии ZigZag (на M5) на графике:
// Наполняем индикаторный буфер ZigZag: на индексы баров с экстремумами
ставим цену, остальные = EMPTY_VALUE.
// (Если используем объекты, то рисуем/обновляем объекты линий между
последовательно идущими экстремумами на M5.)
PlotZigZagLine(pivotsM5);

// 5. Проверка **сигналов входа** (стрелки):
// Условие: M5 только что развернулся (т.е. тренд M5 сменился по сравнению с
предыдущим значением)
if(prevTrendM5 != trendM5) {
    if(trendM5 == "Up") {
        if(trendH1=="Up" && trendM15=="Up") {
            DrawArrow(currentBarIndex, "Buy",
true); // сильный сигнал Buy (яркая стрелка)
            openPositionSide = "Buy"; // считаем, что
открыли покупку
        }
        else if(trendH1=="Up" && trendM15 != "Up") {
            DrawArrow(currentBarIndex, "Buy", false); // ранний сигнал Buy
(полупрозрачная стрелка)
        }
    }
}

```

```

        openPositionSide = "Buy"; // (опционально)
помечаем открытую позицию
    }
}
else if(trendM5 == "Down") {
    if(trendH1=="Down" && trendM15=="Down") {
        DrawArrow(currentBarIndex, "Sell", true); // сильный сигнал
        Sell (яркая стрелка)
        openPositionSide = "Sell";
    }
    else if(trendH1=="Down" && trendM15 != "Down") {
        DrawArrow(currentBarIndex, "Sell",
false); // ранний сигнал Sell (бледная стрелка)
        openPositionSide = "Sell";
    }
}
}
// Функция DrawArrow должна отрисовать объект стрелки нужного цвета/
прозрачности над или под свечой с индексом currentBarIndex.

// 6. Проверка **сигналов выхода** (Exit) после входа и **разворота тренда**
(Reversal):
if(openPositionSide == "Buy") {
    // Выход для Buy: пробитие pivot M5 вниз или разворот M15 вниз
    if(Close_price_M5 < previousPivotM5 || trendM15 == "Down") {

        DrawExitMark(currentBarIndex); // рисуем крестик выхода над текущим баром
        openPositionSide = "None"; // считаем позицию закрытой (сброс
состояния)
    }
} else if(openPositionSide == "Sell") {
    // Выход для Sell: пробитие pivot M5 вверх или разворот M15 вверх
    if(Close_price_M5 > previousPivotM5 || trendM15 == "Up") {
        DrawExitMark(currentBarIndex);
        openPositionSide = "None";
    }
}
// (Здесь previousPivotM5 - уровень pivot M5, актуальный на момент входа или
образовавшийся сразу после входа.)

// Отслеживание разворота глобального тренда (H1):
if(prevTrendH1 != trendH1 && trendH1 != "Flat") {
    if(trendH1 == "Up" && trendM15 == "Up") {
        DrawReversalSign(currentBarIndex, true); // значок разворота вверх
(зеленый)
    }
    else if(trendH1 == "Down" && trendM15 == "Down") {
        DrawReversalSign(currentBarIndex, false); // значок разворота вниз
(красный)
    }
}
}

```

```
// 7. Обновление панели статуса:
string statusText = "H1:" + trendH1 + " M15:" + trendM15 + " M5:" +
trendM5;
ObjectSetString(0, "TrendStatus", OBJPROP_TEXT, statusText);

return rates_total; // завершение расчета индикатора
```

(Примечание: В псевдокоде выше `currentBarIndex` – индекс текущего бара (например, 0 для последнего), `Close_price_TF` – цена закрытия на соответствующем таймфрейме. В реальном коде MQL5 для получения цен старших TF можно использовать `CopyRates` или ту же `CopyBuffer ZigZag`, либо рассчитывать тренд по известному pivot без прямого доступа к Close старшего периода.)

Комментарии к алгоритму: Логика сигналов построена так, как описано: стрелки появляются при смене тренда M5, если это движение согласовано с другими TF; значки выхода – при пробое значимых уровней против позиции; значок разворота – при смене глобального тренда H1. Важно учесть некоторые технические детали при реализации на MQL5:

- **Очистка старых объектов:** чтобы график не перегружался, можно удалять устаревшие стрелки и метки по прошествии какого-то времени или количества баров, либо ограничивать их количество. Например, хранить в массиве имена последних N нарисованных объектов и при появлении новых сигналов удалять самые старые.
- **Перерисовка ZigZag:** ZigZag индикатор имеет свойство перерисовывать последние точки, когда появляется новый экстремум. Это значит, что свежесформированный потенциальный pivot может впоследствии сместиться, если рынок продолжит движение. В нашем алгоритме эта проблема минимизирована тем, что мы берем для анализа предыдущий заверченный экстремум в качестве pivot (а не текущий формирующийся). Тем не менее, при тестировании кода стоит учитывать, что ZigZag меняет последние точки задним числом, и убедиться, что сигналы учитывают **подтвержденные** экстремумы.
- **Координаты объектов по времени и цене:** при рисовании стрелок, крестиков важно указывать правильные координаты **времени** (временная метка бара) и **цены**. Например, `ObjectCreate(..., Time[currentBar], PriceLevel, ...)`. Обычно для стрелки на **покупку** в качестве PriceLevel берут Low бара минус небольшой отступ, чтобы стрелка не перекрывала свечу; для **продажи** – High бара плюс отступ.
- **Прозрачность цветов:** в MQL5 можно задавать цвета с альфа-каналом (прозрачностью). Например, `uint semiColor = ARGB(128, 0, 255, 0);` даст полупрозрачный зеленый (128 – значение непрозрачности из 255). Используя ARGB, можно установить `OBJPROP_COLOR` для стрелки раннего сигнала с ~50% прозрачности. Если по каким-то причинам ARGB недоступен (например, в `OBJ_ARROW`), можно использовать `OBJ_LABEL` с Wingdings, где ARGB точно поддерживается, или альтернативно менять размер/стиль символа для визуального отличия.

Заключение

Сформированный по этому алгоритму индикатор будет в реальном времени отображать на одном графике (например, M5) сразу три уровня **MF-pivot** (с H1, M15, M5), актуальную **линию ZigZag** рабочего таймфрейма, а также динамически появляющиеся **стрелки входа, крестики выхода и значки разворота** – все в полном соответствии с методикой MasterForex-V. Такой инструмент существенно упрощает принятие решений, предоставляя своеобразную «панель

приборов» трейдера – все ключевые сигналы и уровни у него перед глазами. Мульти-таймфреймовый анализ выполняется автоматически, позволяя эффективно оценивать состояние рынка и получать торговые подсказки прямо на рабочем графике.

Используя данный алгоритм в MQL5, вы получите **полностью функциональный индикатор MasterForex-V**. Он поможет видеть согласованное движение рынка на нескольких ТФ, вовремя входить в рынок при подтверждении тренда и выходить при его окончании. При необходимости индикатор можно расширять – добавлять авторские фильтры, звуковые оповещения, либо на его основе создать советника, автоматизировав открытие и закрытие сделок. Такой подход повышает информированность трейдера и уверенность в ежедневной торговле.

Источники:

- MasterForex-V. Описание концепции MF-pivot и мульти-таймфреймового анализа ¹⁰ ¹¹ .
- MQL5 Documentation – **Returning values of ZigZag indicator** (пример использования ZigZag через iCustom и фильтрации значений) ² ¹² .
- MQL5 Articles – **MTF indicators as the technical analysis tool** (принципы построения мульти-тф индикаторов) ¹ .

¹ MTF indicators as the technical analysis tool - MQL5 Articles

<https://www.mql5.com/en/articles/2837>

² ¹² Returning values of ZigZag indicator - Indices - Technical Indicators - MQL5 programming forum

<https://www.mql5.com/en/forum/444601>

³ ⁴ ⁵ ⁶ ⁷ ⁸ ⁹ Алгоритм индикатора MasterForex-V для MetaTrader 5.pdf

<file:///file-HwjSigK8kvQGLkFD6AtQRe>

¹⁰ ¹¹ Pivot Points in Forex - Trading Discussion - BabyPips.com Forum

<https://forums.babypips.com/t/pivot-points-in-forex/180>