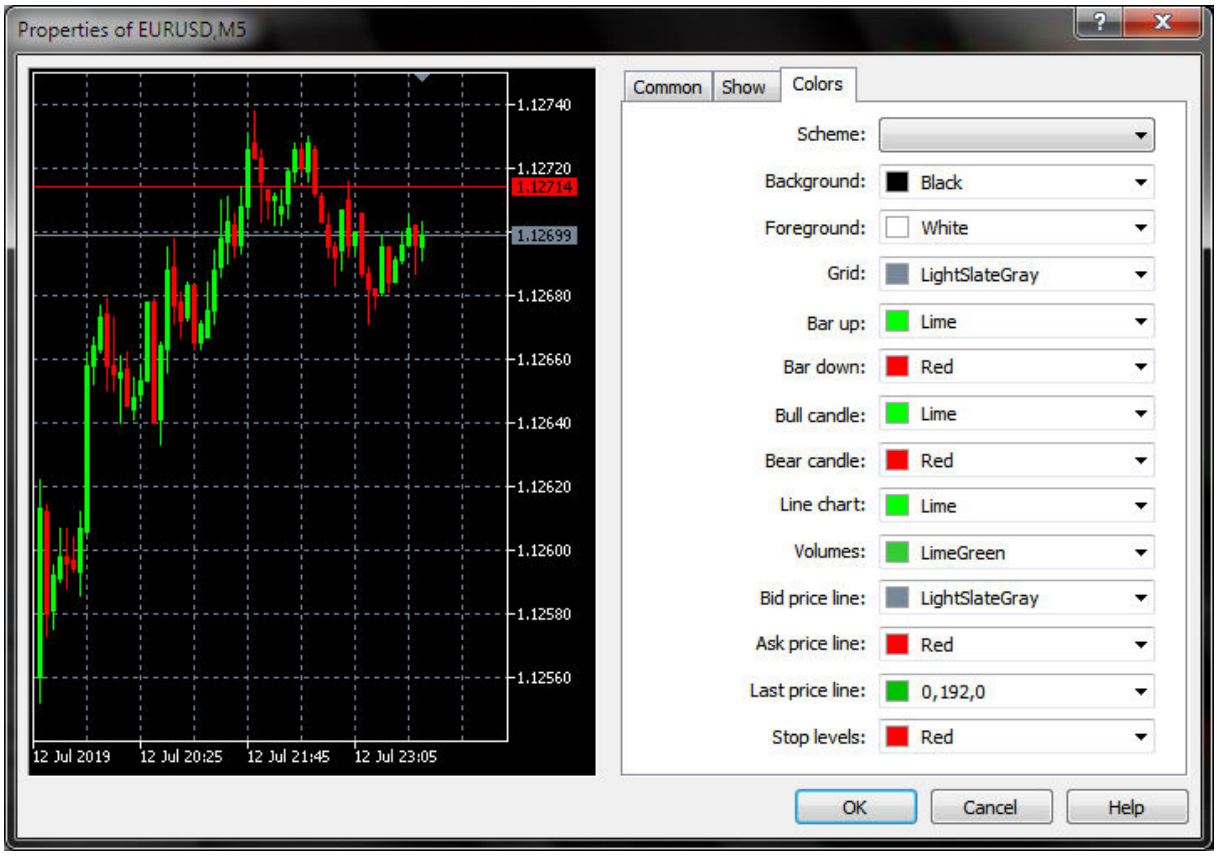
This document contains the description of the workflow of Trading Robot. I would like to call the Trading Robot **"Fibonator".** Icon for the Trading robot is provided in the Order for developing of a Trading Robot(*fibonator_logo.png*). The description of the workflow is separated in 3 stages - The first stage is the preparations before trading, the second stage is the settings for the Trading Robot and the third stage is the trading strategy itself.

STAGE 1 - Prerequisites

On attaching of the Trading Robot to a currency pair it will set the currency pair graphic to candlesticks with color scheme provided bellow.

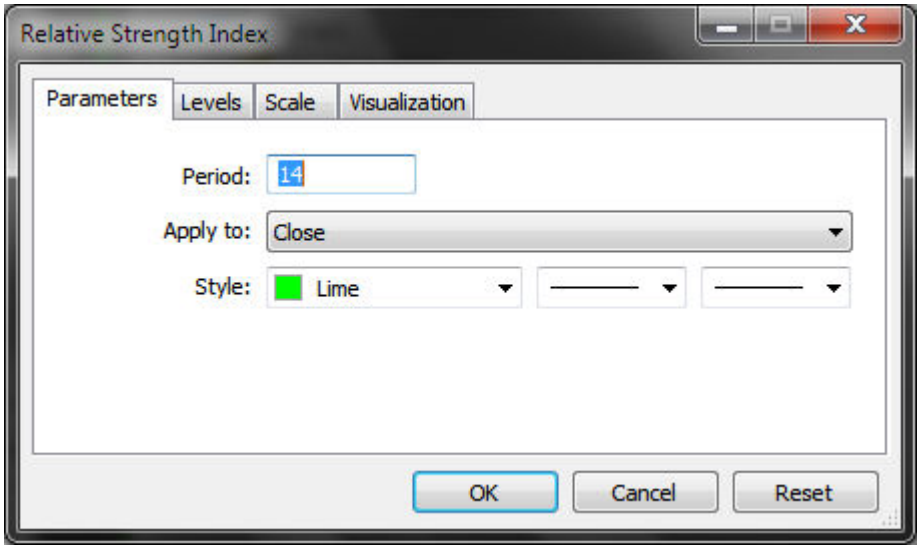
Color scheme

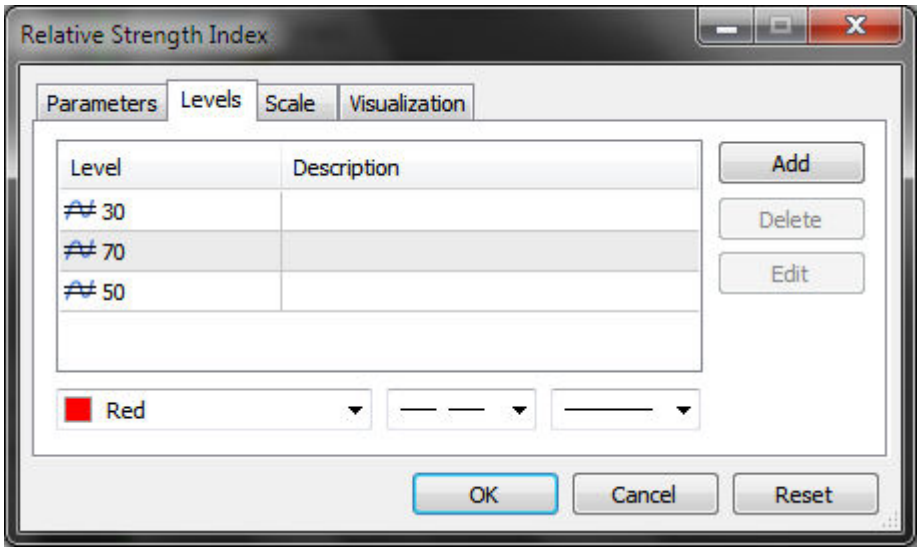


After that it should place the following indicators with specific settings that **<u>will not</u>** be customizable be the user:

1. RSI (Relative Strength Index)

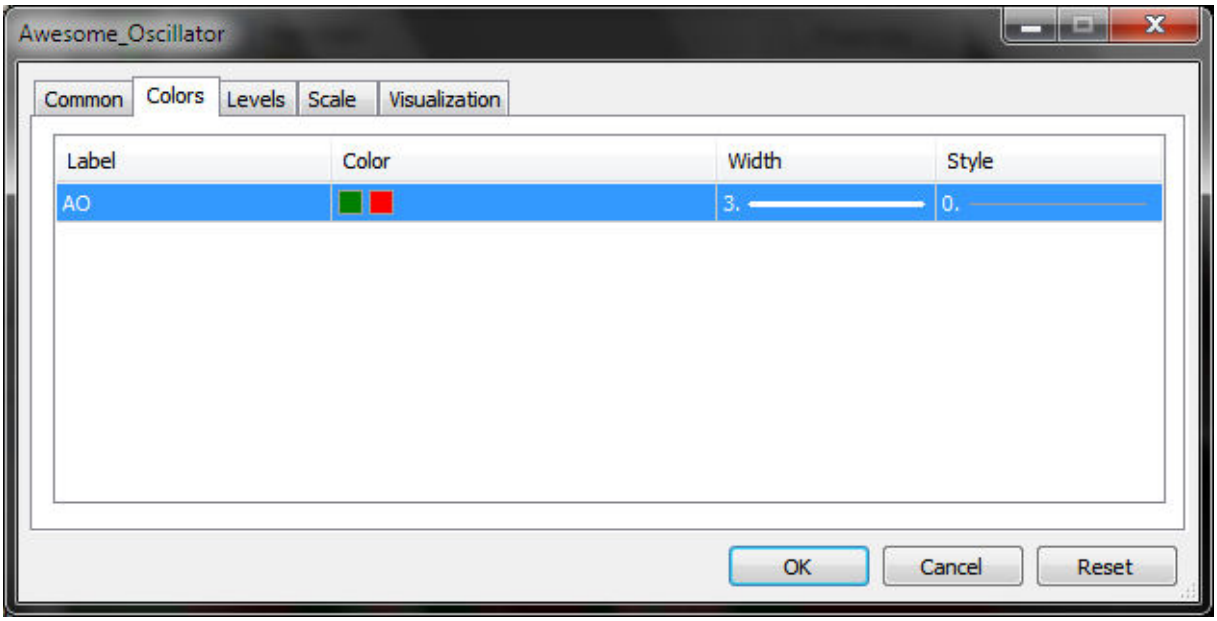
Settings :

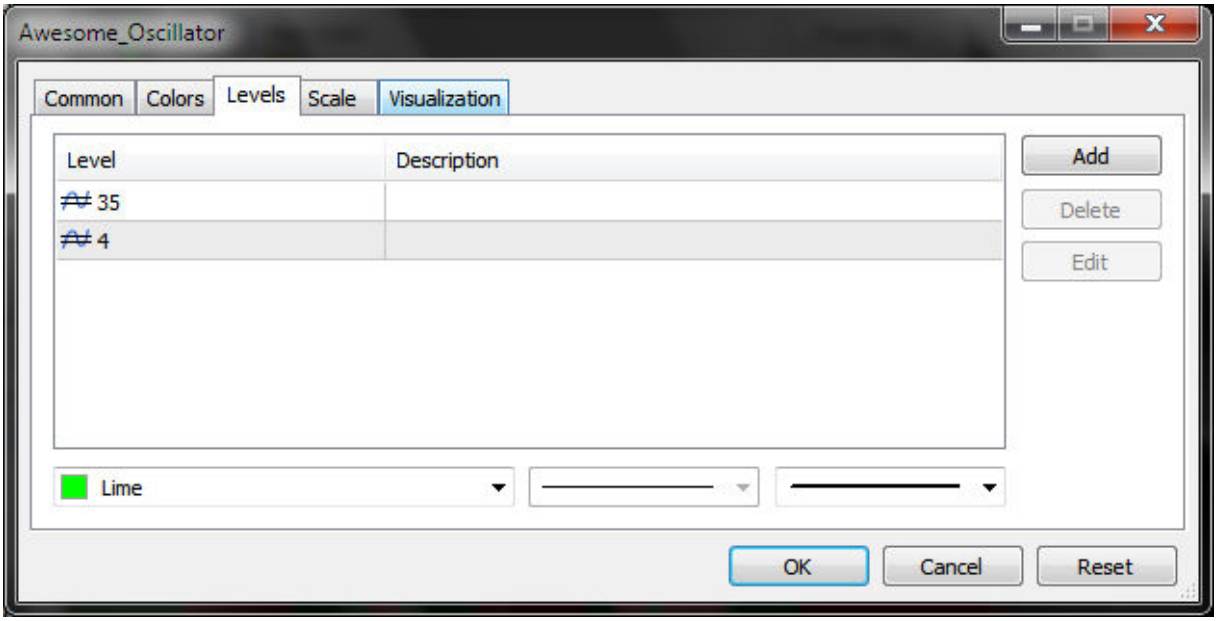




2. Awesome Oscillator

Settings :

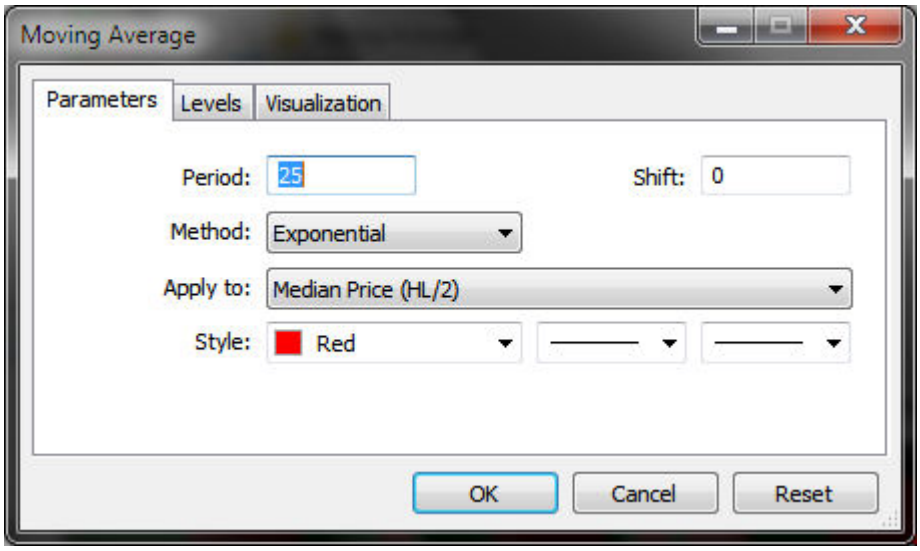




3. Moving Average

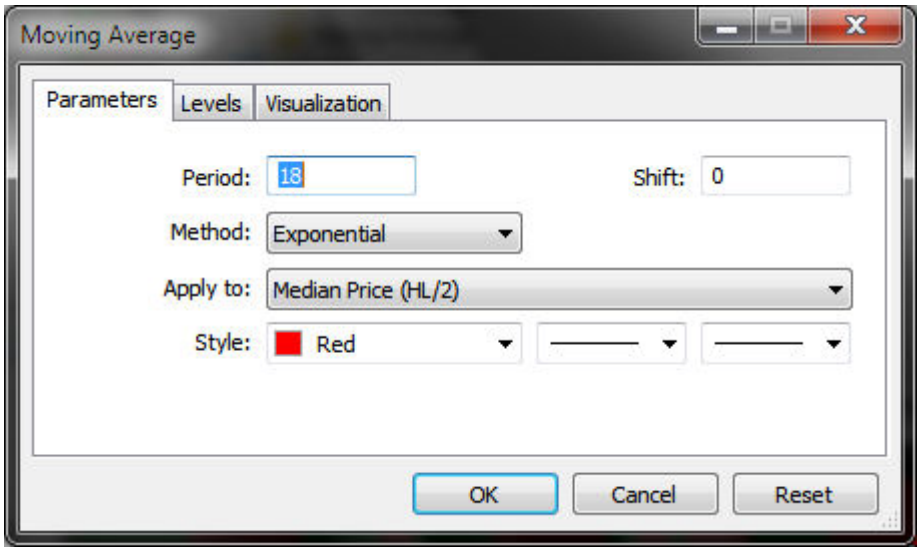
3.A First Moving Average

Settings :



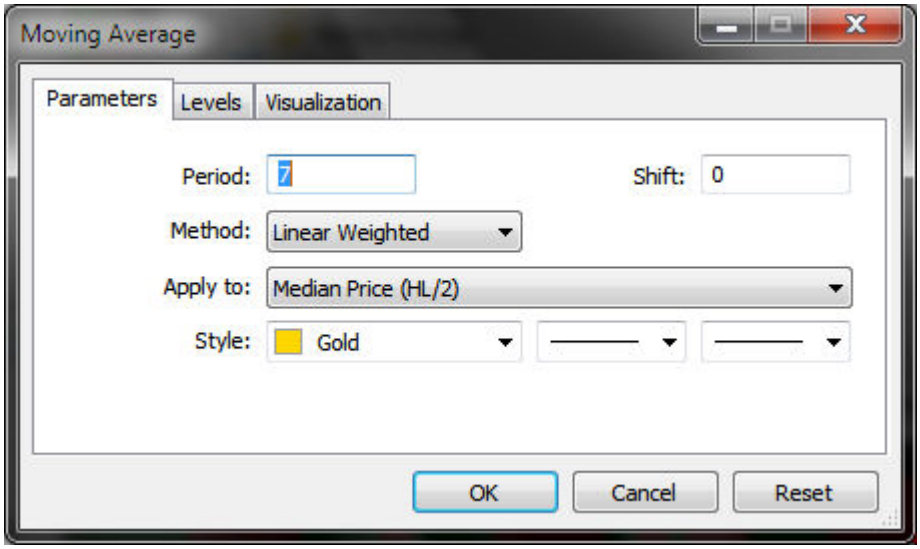
3.B Second Moving Average

Settings :

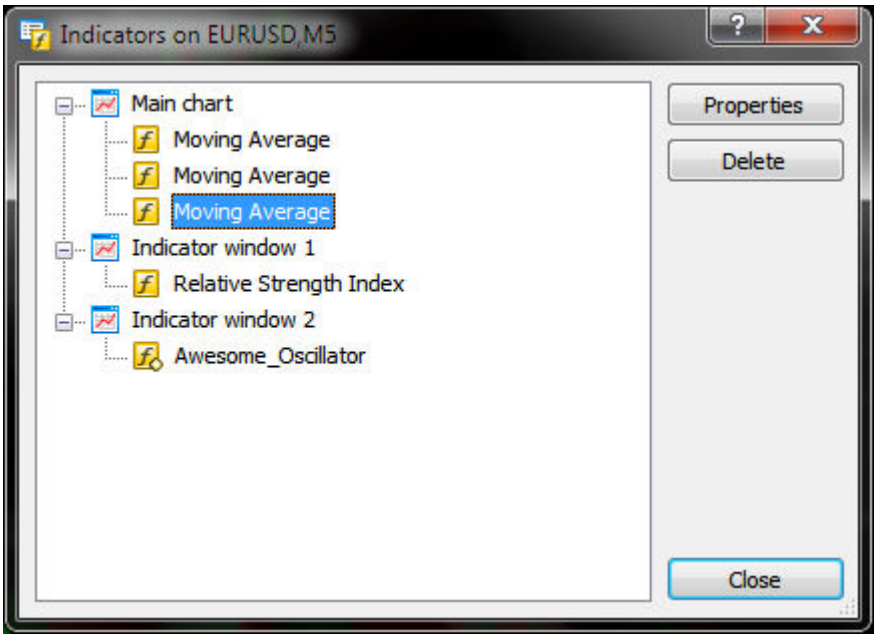


3.C Third Moving Average

Settings :



The final result will be to have all the indicators in this way:



And the data window should look like this(Example) :

STAGE 2 - Settings

Settings for trading strategy of the Trading Robot that **can be customizable** by the user. Every setting is just listed here and each one of it will be explained in detail, further down in this document:

1. Trading LOT size
2. Trading LOT autoincrement
3. Trading LOT autoincrement trigger
4. Autoincrement type
5. Autoincrement LOT value
6. Take profit
7. Stop loss
8. RSI overbought threshold
9. RSI oversold threshold
10. Awesome oscillator detection
11. Fibonacci height
12. Place order level

Explanation of the settings:

1. Trading LOT size - this will be the size of LOT in which the Trading Robot will place for each position.

Field type: Number field
Default value: 0.01

2. Trading LOT autoincrement - This setting will be only **ON** and **OFF**. The purpose of it will be to increment the size of the lot when the account amount of money grows(**the account balance, not equity**). The default setting will be **OFF** and when set to **OFF** - settings with number 3, 4 and 5 will be inactive. When the setting is **ON** the settings with number 3, 4 and 5 will be active.

Field type: Dropdown select
Default value: Off

3. Trading LOT autoincrement trigger - This will be a percentage of the current account amount balance(not equity) - when the value reaches that amount - autoincrement is triggered.

<u>Trading LOT autoincrement trigger Example:</u>

I have account balance of 1000$ if the **Trading LOT autoincrement trigger** is set to 50%

when my account balance grows to 1500$ it will trigger LOT autoincrement
when my account balance grows to 2250$ it will trigger LOT autoincrement
when my account balance grows to 3375$ it will trigger LOT autoincrement

Field type: Number field
Default value: 50%

4. Autoincrement type - This will have two options **- Fixed rate** or **Multiplier**.

<u>**Fixed rate** example:</u>

I have initial LOT set to 0.01 and **Autoincrement LOT value** is set to 0.01. When **Trading LOT autoincrement trigger** occurs the LOT for the next position will be 0.02.
I have initial LOT set to 0.02 and **Autoincrement LOT value** is set to 0.01. When **Trading LOT autoincrement trigger** occurs the LOT for the next position will be 0.03.
I have initial LOT set to 0.05 and **Autoincrement LOT value** is set to 0.01. When **Trading LOT autoincrement trigger** occurs the LOT for the next position will be 0.06.

Multiplier example:

I have initial LOT set to 0.01 and **Autoincrement LOT value** is set to 2. When **Trading LOT autoincrement trigger** occurs the LOT for the next position will be 0.02.
I have initial LOT set to 0.02 and **Autoincrement LOT value** is set to 2. When **Trading LOT autoincrement trigger** occurs the LOT for the next position will be 0.04.
I have initial LOT set to 0.02 and **Autoincrement LOT value** is set to 3. When **Trading LOT autoincrement trigger** occurs the LOT for the next position will be 0.06.

Field type: Dropdown select
Default value: Multiplier

5. Autoincrement LOT value - The purpose and usage of this field value is described above in point **4 Autoincrement type.**

Field type: Number field
Default value: 2

6. Take profit - This the desired profit size in pips.

Take profit example:

The Trading Robot opens **BUY** position at 1.12600 and the **Take Profit** is set to 40. Then the take profit for the position is 1.12640

The Trading Robot opens **SELL** position at 1.12600 and the **Take Profit** is set to 40. Then the take profit for the position is 1.12560

Field type: Number field
Default value: 40

7. Stop loss - The desired stop loss size in pips.

Stop loss example:

The Trading Robot opens **BUY** position at 1.12700 and the **Stop loss** is set to 150. Then the stop loss for the position is 1.12550

The Trading Robot opens **SELL** position at 1.12600 and the **Stop loss** is set to 150. Then the stop loss for the position is 1.12750

Field type: Number field
Default value: 150

8. RSI overbought threshold

<u>This is applied only on BUY positions!</u>

This is one of the factors which is used in the trading strategy, lets call it a **"Decision factor"** and used by the Trading Robot to decide how to proceed in cases of **"Place order event"** occurs (which will be described later in Stage 3 - Trading Strategy). This setting operates in this way: we match the **"Place order event"** with the current RSI value and see if this value is within the RSI overbought threshold range. RSI overbought threshold range is implied by the value of **RSI overbought threshold** added after the 50% line of RSI (positive overhead).

Example:

We have **RSI overbought threshold** set to 20 - this means we have RSI overbought threshold range of 50-70, we match the **"Place order event"** with the RSI value at the moment and if the RSI value is 60 (within the range) the setting will return TRUE.

We have **RSI overbought threshold** set to 25 - this means we have RSI overbought threshold range of 50-75, we match the **"Place order event"** with the RSI value at the moment and if the RSI value is 80 (out of the range) the setting will return FALSE.

Field type: Number field
Default value: 20

9. RSI oversold threshold

<u>This is applied only on SELL positions!</u>

This is one of the factors which is used in the trading strategy, lets call it a **"Decision factor"** and used by the Trading Robot to decide how to proceed in cases of **"Place order event"** occurs (which will be described later in Stage 3 - Trading Strategy). This setting operates in this way: we match the **"Place order event"** with the current RSI value and see if this value is within the RSI oversold threshold range. RSI oversold threshold range is implied by the value of **RSI oversold threshold** added bellow the 50% line of RSI (negative downhead).

Example:

We have **RSI oversold threshold** set to 20 - this means we have RSI oversold threshold range of 30-50, we match the **"Place order event"** with the RSI value at the moment and if the RSI value is 40 (within the range) the setting will return TRUE.

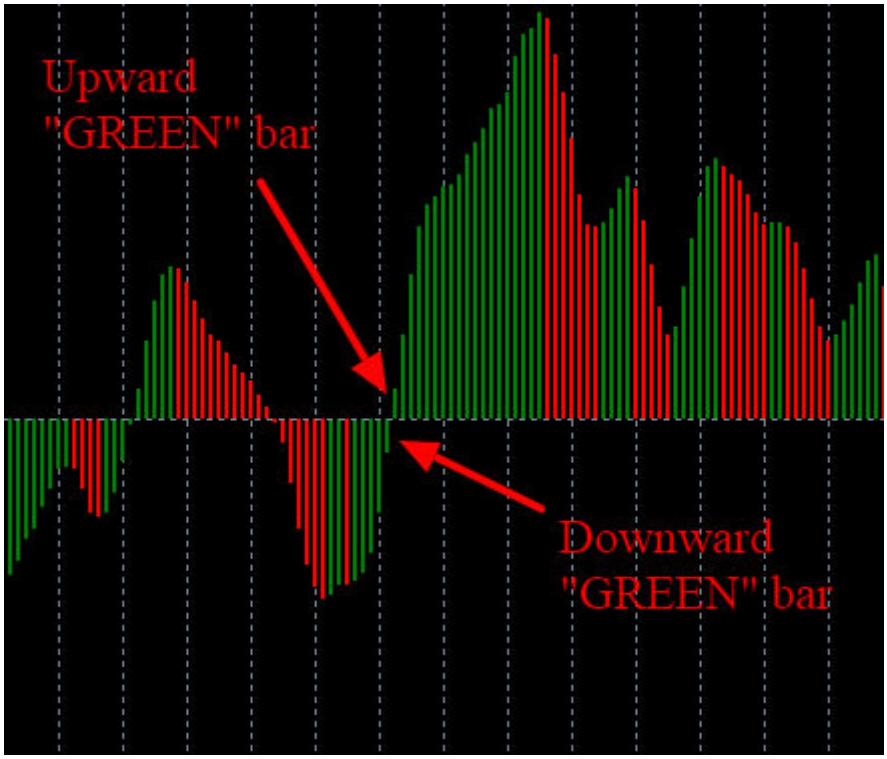
We have **RSI oversold threshold** set to 25 - this means we have RSI overbought threshold range of 25-50, we match the **"Place order event"** with the RSI value at the moment and if the RSI value is 23 (out of the range) the setting will return FALSE.

Field type: Number field
Default value: 20

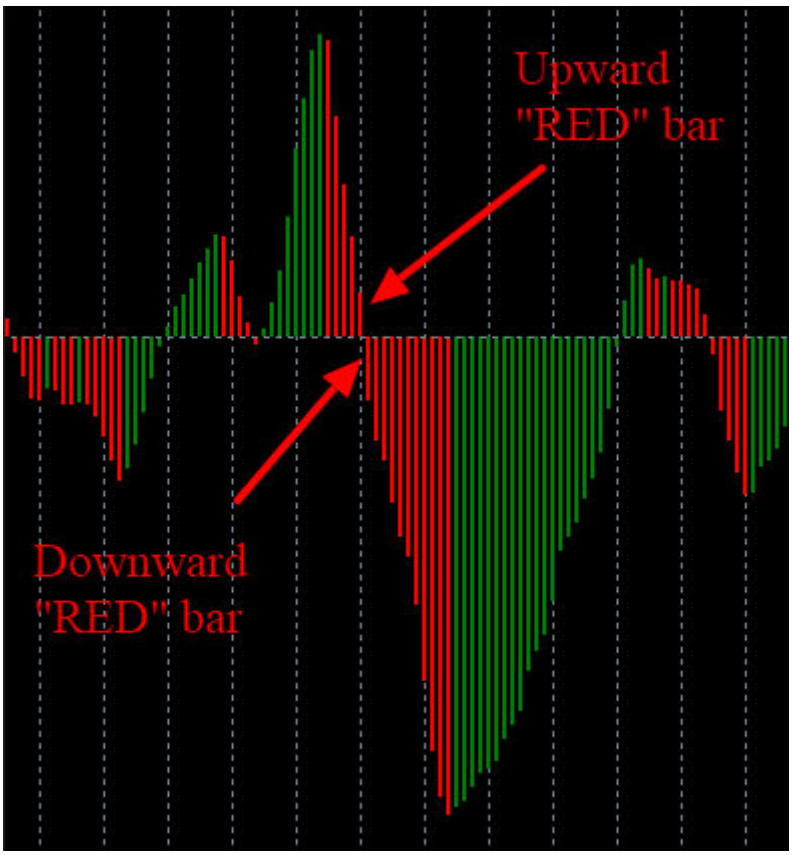
10. Awesome oscillator detection - This will have two settings **Neutral** and **Direction**
This is one of the factors which is used in the trading strategy, lets call it a **"Decision factor"** and used by the Trading Robot to decide how to proceed in cases of **"Place order event"** occurs (which will be described later in Stage 3 - Trading Strategy).

Neutral setting explained:

When a **"Place order event"** occurs for **BUY** position we check if the bar is "GREEN" at this time and the setting will return TRUE whether the bar is upwards or downwards.



When a **"Place order event"** occurs for **SELL** position we check if the bar is "RED" at this time and the setting will return TRUE whether the bar is upwards or downwards.



Direction setting explained:

When a **"Place order event"** occurs for **BUY** position we check if the bar is "GREEN" at this time and the setting will return TRUE only if the bar is upwards.

When a **"Place order event"** occurs for **SELL** position we check if the bar is "RED" at this time and the setting will return TRUE only if the bar is downwards.

Field type: Dropdown select
Default value: Neutral

11. Fibonacci height - this will be the size of the Fibonacci retracement object that is placed when **"Place order event"** occurs. I am not 100% for here with the terminology but lets call it "Height" and it will be the value shown on screenshot bellow. This setting will be customizable by the user, but the other setting "Width" should be hidden an with fixed value of 40. The "Start point" will be the **"Place order event"** which will be described later in Stage 3 - Trading strategy**.**

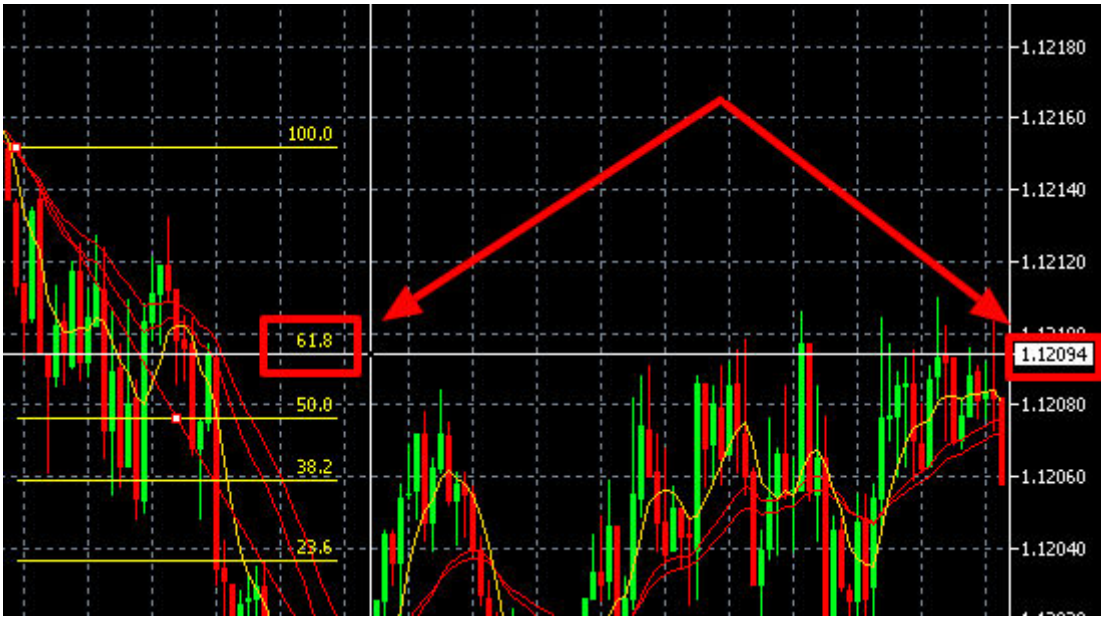
NOTE: The screenshot above shows Fibonacci retracement for BUY position, for SELL it should be mirrored!

Field type: Number field
Default value: 150

12. Place order level - this setting will control on what level of the Fibonacci retracement the pending order will be placed. Example:

On level 50 it will place pending order at 1.12076

On level 61.8 it will place pending order at 1.12094



Field type: Number field
Default value: 50

STAGE 3 - Trading Strategy

The trading strategy generally is based on this flow:

1. We detect the **"Place order event"**
2. We check the **"Decision factors"** if they are TRUE.
3. If we have conditions for trade - the Trading Robot places pending order based on the Trend direction(see Note 1) **"Place order level ", "Trading LOT size", "Take profit"** and **"Stop loss"** settings.

Note 1: If the Gold MA line is crossing the other two Red MA lines from bellow this means the pending order will be BUY, if it is crossing from above the pending order will be SELL.

The **"Place order event"** is the event in which **the GOLD colored MA line crosses the other two RED colored MA lines**. The direction of crossing determines the order type - BUY or SELL. If the line crosses from bellow it is a BUY, if the line crosses from above it is a SELL. The **"Place order event"** is the point where we place the **"Start point"** of the Fibonacci retracement if the other two **"Decision factors"** are met.

The trading robot will not place Fibonacci retracement if the two "Decision factors" do not return TRUE!

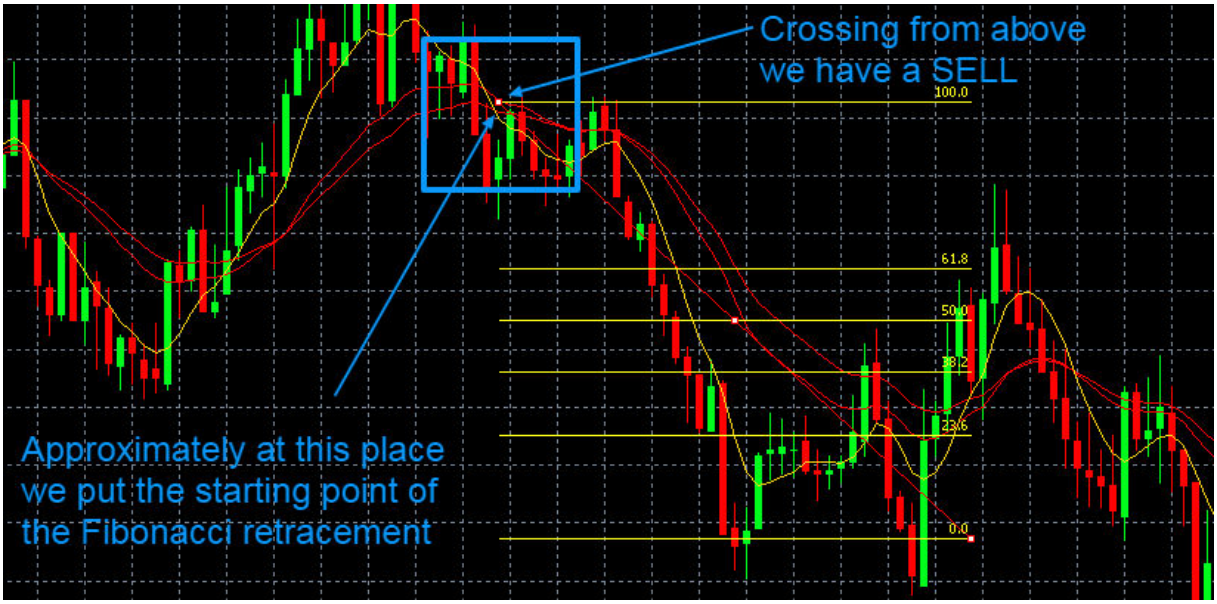
*If we have pending order left from the previous **"Place order event"** when the next **"Place order event"** occurs and the Trading Robot places another pending order and the new order will cancel the previous one! Consider this as sort of clean start. Example:*
*The Trading robot places pending order when conditions are met, by the time of waiting the order to be executed, the Trading Robot "decides" to place another order and upon placing the new pending order the previous one is canceled. **This apply only for pending orders not executed ones!***

Example of **"Place order event"** for BUY :

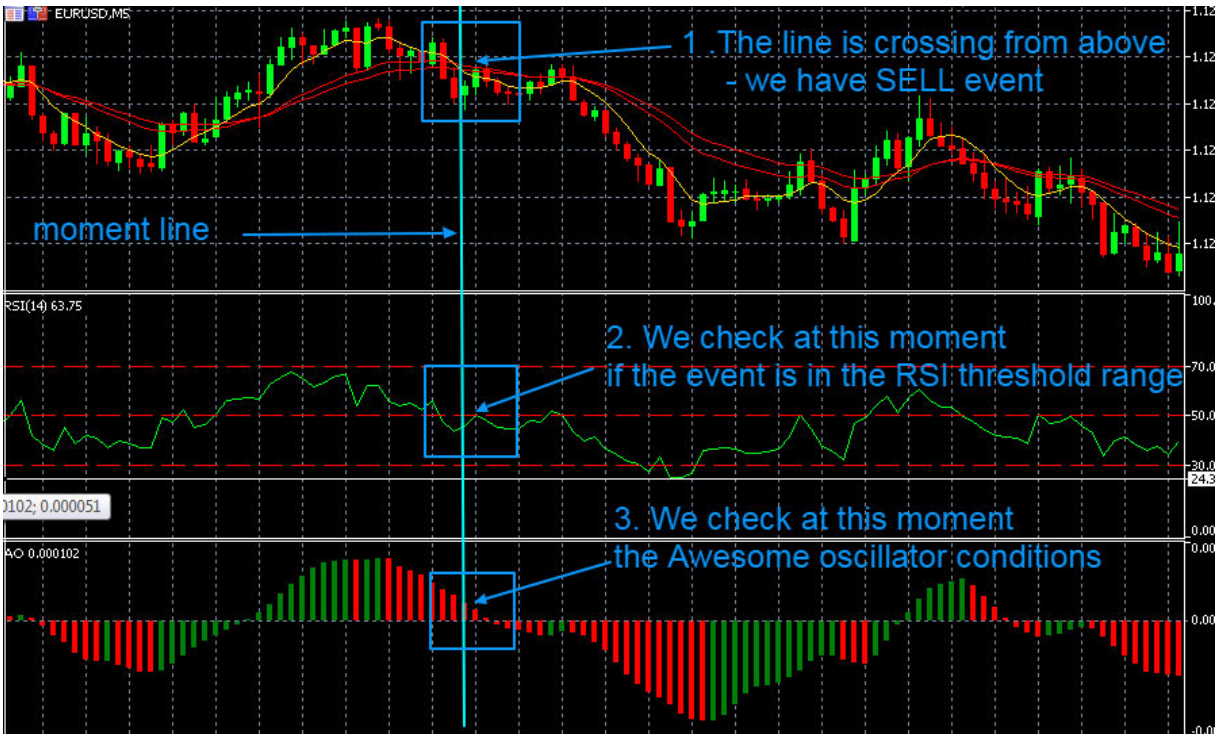
Example of **"Place order event"** for SELL :

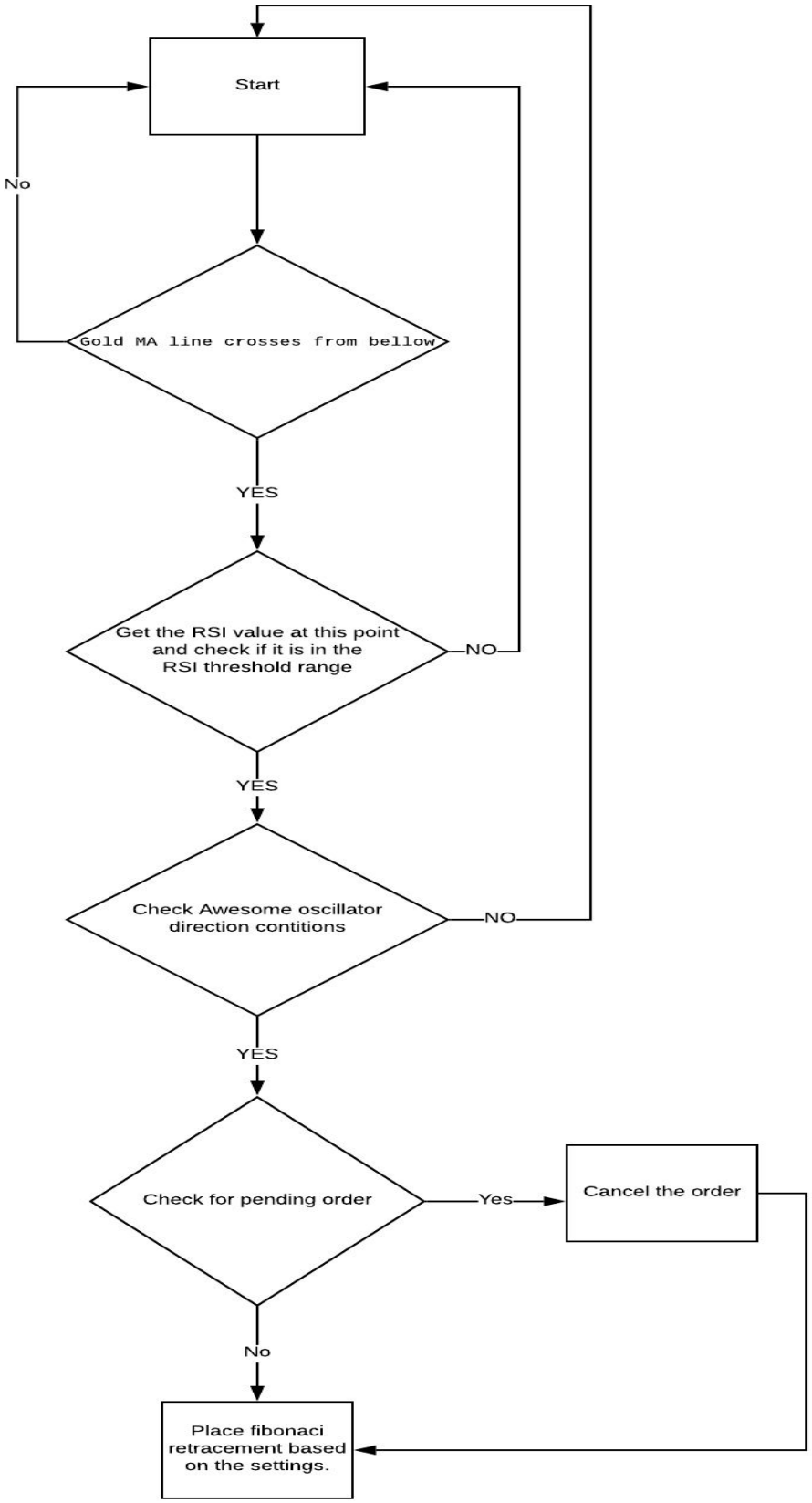
The points shown in the screenshot above is where the Gold MA line crosses the other two red MA lines will be the "Starting point" of the fibonacci retracement. I know it is a tricky to determine this but that is left to you to use your "magical" programming skills. :)

Another example of **"Place order event"** for SELL if other Decision factors are met:



Another example of the workflow:



The moment where the Gold MA line crosses the other two red MA lines is the line with TEAL color(moment line) and right at this the moment we check if the other two conditions are met. And if everything is OK we place a Fibonacci retracement.

Example trade logic for BUY: