Non-standard timeframe indicator for МТ5

**Description**

The indicator converts the standard timeframe price into blocks of N points. It should work as quickly and efficiently as possible. It is to be used within the EA for real accounts. The code is to be transferred to the EA. The indicator is needed for visualization. It should be displayed in the price window and a separate window. All calculations are based only on bar closing prices. Recalculation is performed once with each new bar. Only the necessary minimum should be recalculated. If one block to the right should be added to the existing number of blocks, there is no need to recalculate the already formed blocks. The code should contain no errors. It should be understandable by a third-party programmer and easily scalable/modifiable. All data used to build blocks and displayed in the data window should be available for reading by the EA. The indicator should work correctly on any instrument available in the terminal, including all currency pairs, CFDs, stocks, futures, commodities and cryptocurrencies.

**Options**

- Fix time: yes/no

- Bar date and time (-1): (set in the date and time format up to minute)

- Block size: (set in price format, for example 0.00001)

- Number of blocks to build: (integer from 1 to 1,000,000)

- Number of timeframes: (1-10,000, integers)

- Ratio of adding the next timeframe: 0 - not used (set in the price format, for example 0.00001)

- Next timeframe multiplication ratio: (fractional, 0.000001 - 100)

- rounding precision (for example, 0.00001)

- Calculate the excess percentage: yes/no

- Build blocks in the chart window: yes/no

- Build in a separate window: yes/no

- Color of bullish block lines

- Color of bearish block lines

**1. Block forming algorithm**

The indicator is built in a separate window or in a chart window depending on the settings. If **"build blocks in the chart window" = yes** and **"build in a separate window" = yes**, then the indicator is constructed in both locations. If none are selected, it is not constructed anywhere but calculations are performed and all the values are written to the appropriate arrays. These modes are necessary to be used in the EA to check the indicator and EA. Disabling visualization should accelerate the system operation. Construction is performed from left to right, i.e. the first construction and reference point is Close -1. Zero bar is not used. Only Close values, as well as the difference between Close values of neighboring bars are used for construction. The number of points the price should pass vertically to build one block is set in the settings. The block features four parameters: Open, High, Low and Close. To avoid confusion with bar prices, I will name them BOpen, BHigh, BLow and BClose. I will describe the construction algorithm briefly and in details.

You can take the block forming algorithm from the description or compile it on your own. The most important thing is to obtain the correct result.

**Briefly**: We take the difference between Close(-1) and Close(-2) bar prices and check how many whole blocks fit there. If 1 or more, then build them. Next, take the difference between Close(-1) and Close(-3) and check the number of whole blocks fitting the range. If the number of whole blocks exceeds the previous one, the missing blocks are built. If the number is equal to the previous one, then nothing is done and the next action is performed. If the number is less than the one during the previous iteration, the blocks in the opposite direction are built. This continues till the necessary number of blocks is constructed. We take Close(-1) and Close(-4) values etc. Open price of each new block is calculated from the Close price of the bar (-1). Closing price of each block = opening price of the previous one (from right to left) except for the first block. Closing price of the first block = bar closing price (-1). This goes on till the necessary number of blocks specified in the settings is built. Each block has High and Low values. These are price highs and lows that were during the block formation interval. They are built based on High and Low of real bars. After building the necessary number of blocks, the algorithm waits for the new bar to close and the procedure is repeated if **"fix time"** = no. If **"fix time"** = yes, built blocks are not recalculated after a new bar but instead continue to be built to the right of the closing level of the bar (-1), on which the time was fixed. The next appearing bars appear as (+1), (+2), while the new blocks will be built first from the difference between Close(-1) and Close(+1), then Close(-1) and Close(-2) as long as new bars appear.

**Detailed description:**

BOpen, BHigh, BLow, BClose – block construction prices

BOpen(-1), BHigh(-1), Blow(-1), BClose(-1) – prices of the last block (to the right) formed, the zero block is being formed as on the bars.

Close (0) – price of closing the current formed bar

Close(-1) – price of closing the last formed bar

Close (-2) – price of closing the penultimate formed bar

Close(-1) –Close (-2) = number of points.

BS – block size from settings

P1 – difference between close prices of the first two candles

P2 … Pn – difference between closing prices of each subsequent pair of candles.

For example, Р2= Close(-2) –Close (-3) or Рn= Close(-n) –Close (-(n-1)).

**Block size (BS)** is taken from the settings, **Close(-1) –Close (-2) = P1 is calculated next. If P1/BS>=1,** we need to build the first bullish block, the values are calculated for that

BOpen(-1) and BClose(-1). BСlose(-1) = Close(-1), Bopen(-1) = Close(-1)-BS. Next, the high and low wicks are assigned to the block. These are maximum and minimum prices at the time of the block formation. After building the block, **P1/BS>=2 is checked.** If yes, yet another block is built featuring BСlose(-2) = Bopen(-1) and Bopen(-2) = Close(-1)-2\*BS. High and low wicks are assigned. This continues until the answer is no.

If the 'no' answer is received immediately after the **P1/BS>=1** condition, the **If P1/BS<=-1 condition is checked.** If yes, we need to build the first falling block. To achieve this, BСlose(-1) = Close(-1), BOpen(-1) = Close(-1)+BS. Next, high and low wicks are assigned to the block. The wicks correspond to the maximum and minimum price values within the block formation period. Next, the **P1/BS<=-2** value is checked. If yes, yet another bearish block is built. It features Bclose(-2)=BOpen(-1) and BOpen(-2)= Close(-1)+2\*BS. High and Low wicks are assigned. This continues until the answer is no.

If 'no' answer is received after **P1/BS>=1** and after **P1/BS<=-1**, the (P1+P2)/BS value is taken. The condition for building a bullish block is checked in a similar manner (**P1+P2)/BS>=1** . If it is not built, the condition for building a bearish block is checked (**P1+P2)/BS<=-1.** If 'yes' is received in one of the conditions, the block, on which the answer was received, is built and the condition for building yet another block is checked as described above. If after checking two conditions, the answer is no, then one more P is added and the following condition is checked: (**P1+P2+Pn)/BS>=1** and(**P1+P2+Pn)/BS<=-1**. The cycle continues till the block is built. Everything described above is shown in Figure 1.

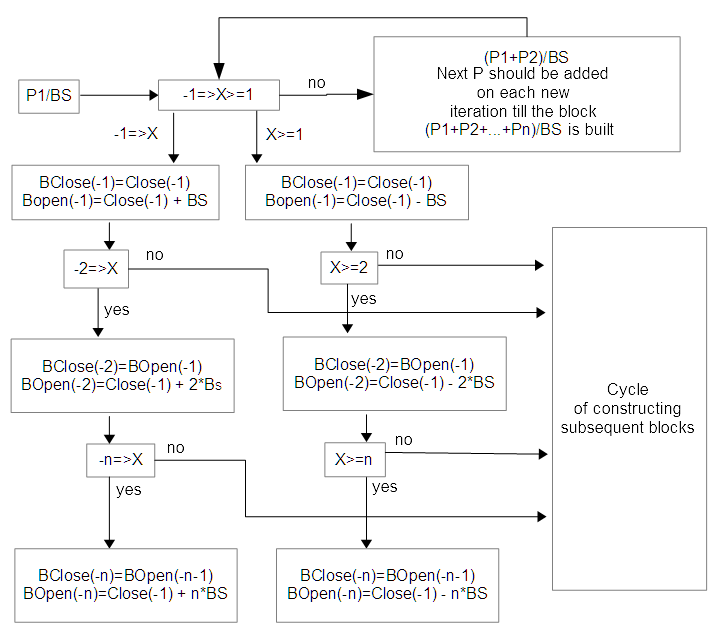


Figure 1. The algorithm of building the first blocks

Suppose that the bullish block was built and 'no' was obtained on the X>=2 condition. Then we need to move on to the next loop shown in Figure 2. To do this, add **P** to the equation where the 'no' answer was obtained. Suppose that the equation was (P1+P2)/BS, then we need to use the equation **(P1+P2+P3)/BS to move on to the next block construction cycle.** Here we need to check the 0=>X condition. If 'yes', move on to constructing the falling block. For that, we need to calculate open and close prices

BСlose(-2)=BOpen(-1), BOpen(-2)=Close(-1)+0\*BS. Next, check the condition -1=>X. If 'yes', build the next falling block using the equations BСlose(-3)=BOpen(-2) and BOpen(-3)=Close(-1)+1\*BS.

If the answer is 'no' when checking the condition 0=>X, the Х>=2 condition is checked. If 'yes', the growing block is built using the BСlose(-2)=BOpen(-1) and BOpen(-2)=Close(-1)-2\*BS equations. The Х>=3 condition is checked next. If 'yes', the next growing block is built using the BСlose(-3)=BOpen(-2) and BOpen(-3)=Close(-2)-3\*BS equations.

As soon as we get 'no', move on to the next cycle of adding P, which works similarly. The cycle is repeated till the number of blocks specified in the settings is built**.**

If 'no' is obtained at the very start of the cycle where **(P1+P2+P3)/BS** was checked, one P is added. The equation becomes **(P1+P2+P3+P4)/BS** and 0=>X is checked again. This is repeated till the 0=>X check yields 'yes' or X>=2. This is followed by block construction and further checks according to the algorithm.

Figure 2 shows two examples. The one to the left demonstrates the first growing block and the 'no' answer obtained when checking the X>=2 condition. Now we need to check the Х<=0 or X>=2 conditions. If the first built block was falling, then we would have to check the X<=-2 or X>=0 condition in the right example in Figure 2. Similar diagrams are built further on depending on which block was built previously. The diagrams reflect the essence of building blocks.

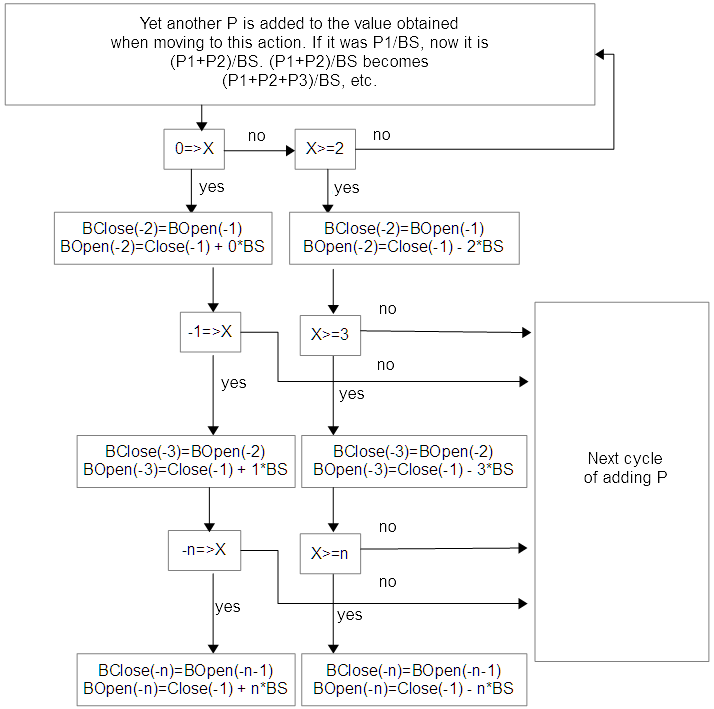
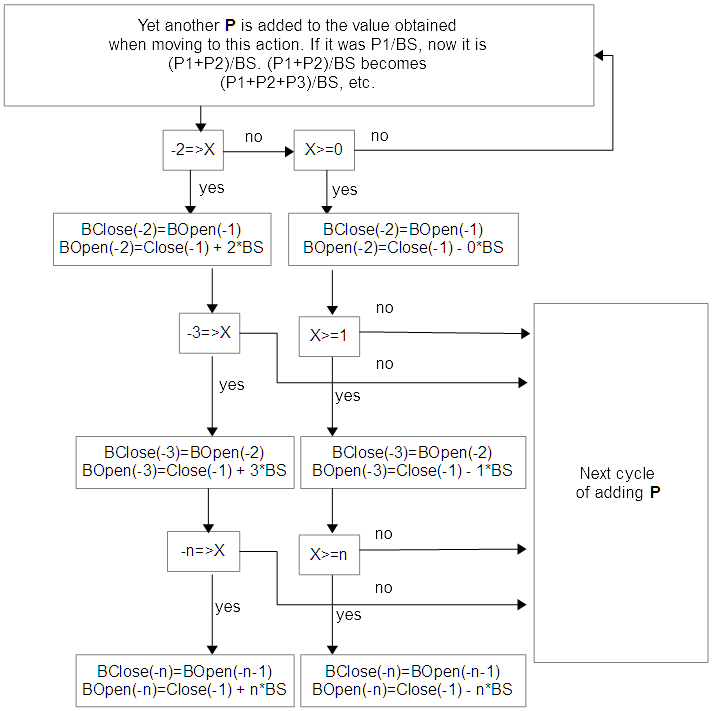
 

Figure 2. The cycle of building the next block

Example:

On the chart, this looks as in Figure 3. The bars are numbered for clarity. The blocks are displayed on the chart and in the indicator window. The wicks are applied to the blocks. I have not managed to apply them to the blocks shown in the indicator window, but they are needed there as well.

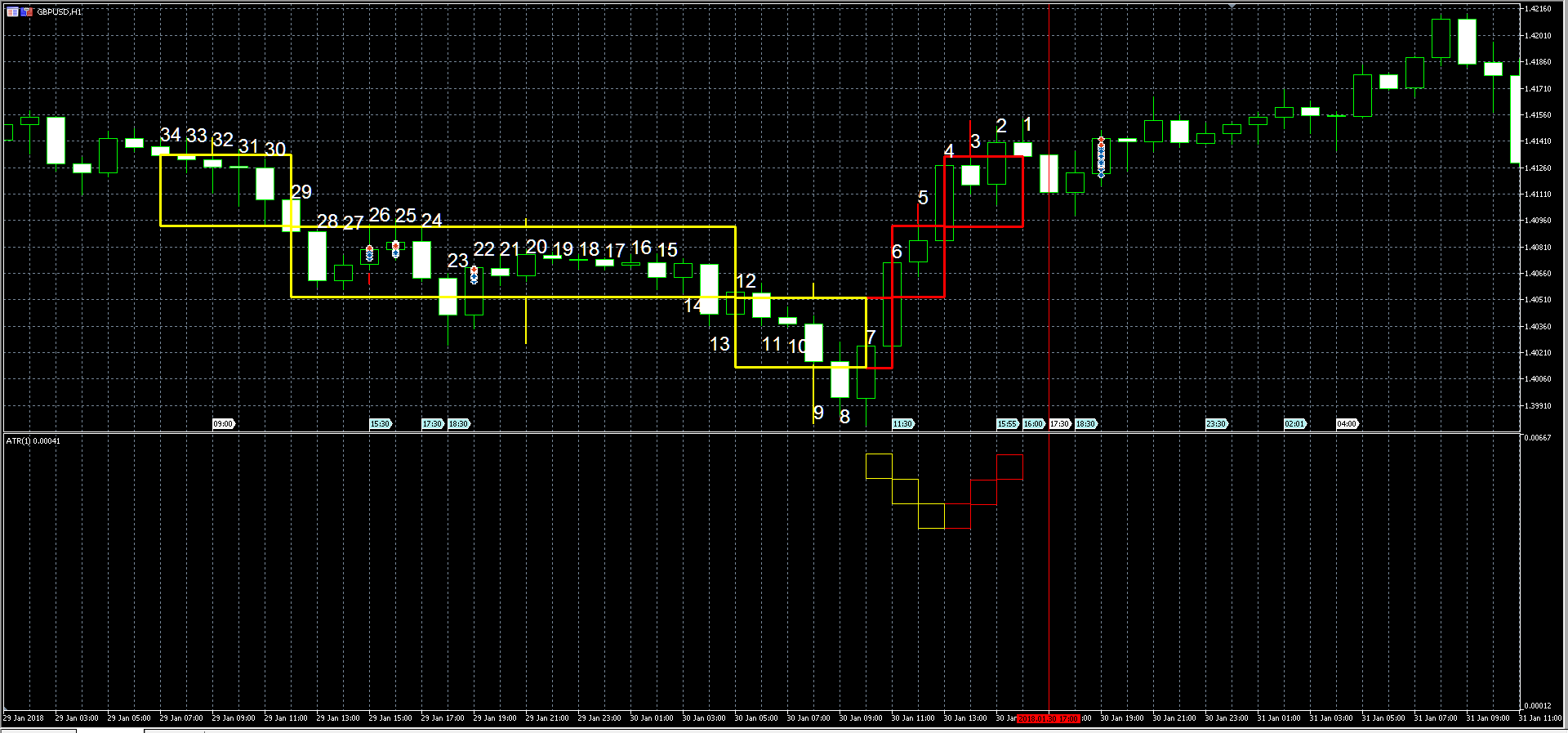


Figure 3. The indicator on the chart

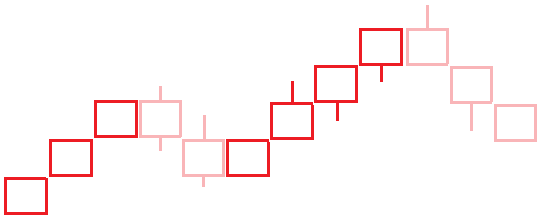


Figure 4. Building blocks

**2. Number of timeframes**

More than one timeframe (blocks of different sizes) can be built on the chart at the same time. Their number is set in the **Number of timeframes setting.** The number range is 1-10,000. The block size of the next timeframe is increased relative to the previous timeframe by one of the values selected in the settings.

A) **ratio of adding the next timeframe**. Set the number of points that should be added to the block size of the current timeframe to get the next one. For example: blocks of the first timeframe 0.00100 points, adding step, 0.00010 points. Thus, the second timeframe block size is 0.00100+0.00010=0.00110, the third one is 0.00110+0.00010=0.00120 etc. Rounding accuracy is set in the "**rounding accuracy**" setting and is written in the price format.

B) **multiplication ratio of the next timeframe** . The multiplier of the block size of the next timeframe is set relative to the current one. For example: multiplier 1.1, first timeframe size 0.00100, then the size of the second timeframe is 0.00100\*1.1=0.00110. The value is rounded up to the minimum price step. If the minimum price step is 0.00001, then it is rounded to 0.00001. The already rounded number is multiplied afterwards. For the third one 0.00110\*1.1=0.00121, etc.

Figure 5 shows several timeframes on one chart. We can see that they all start from the same point.

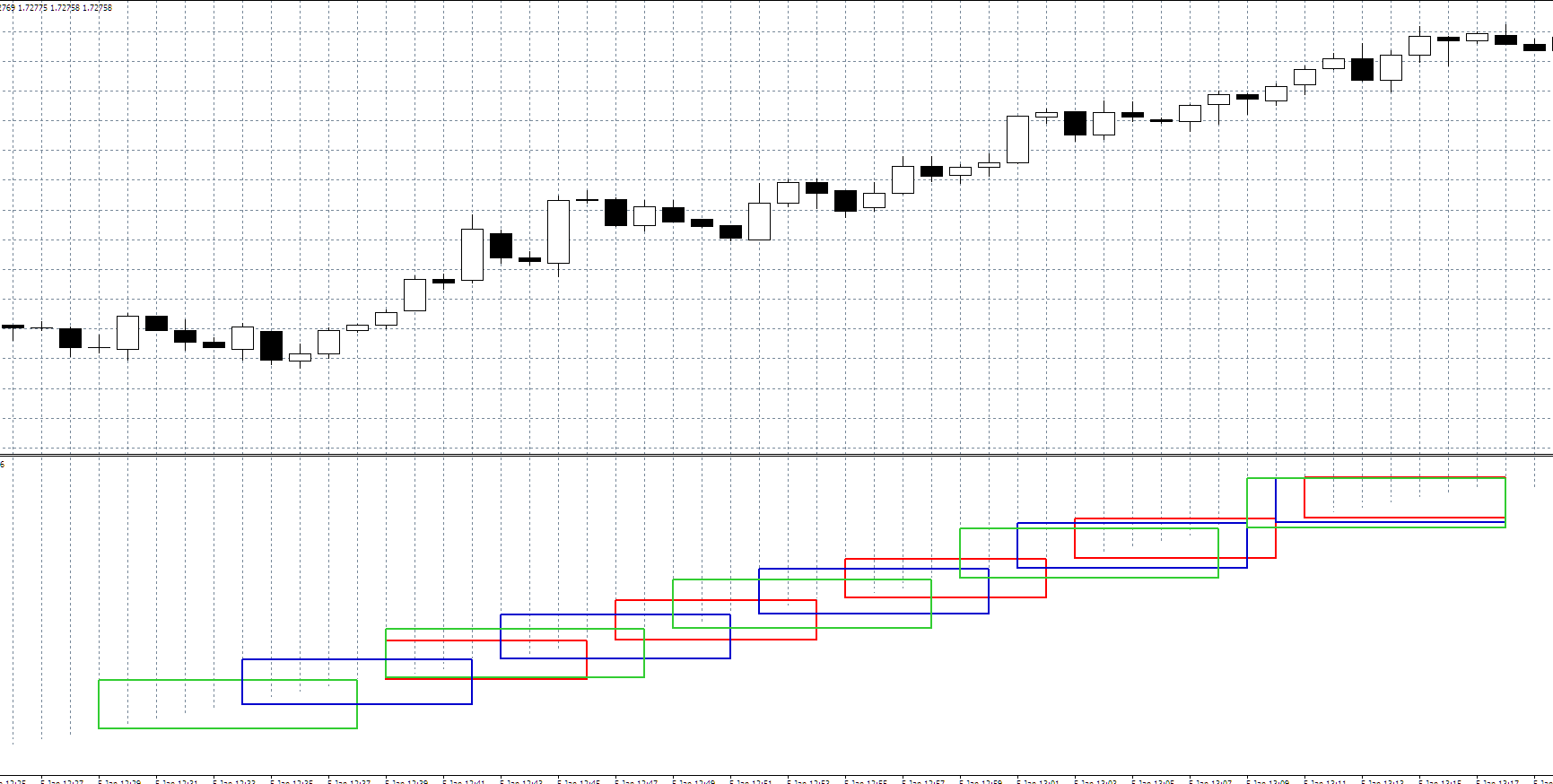


Figure 5. Several timeframes on one chart

**3. Calculate the percentage of growing blocks**

On each built timeframe, we need to calculate the percentage of growing blocks and display it in the data window for each timeframe. The percentage of growing blocks is calculated as (the number of growing blocks)/(total number of blocks)\*100. If "**fix time"=no,** the percentage of growing blocks is calculated only for the number of blocks specified in "**number of blocks for construction**" as described above and is recalculated at each new bar. If "**fix time"=yes,** the percentage of growing blocks is calculated for two cases and, therefore, two values of the growing blocks percentage are displayed in the data window. The first case is calculated for blocks before the bar (-1), while the second one is calculated for the total number of blocks on the current timeframe. Working with blocks after the bar (-1) is described in point 4. If we use an example provided in Figure 6, the initial percentage of growing blocks = 3/6\*100=50%, while the current excess percentage = 6/10\*100=60%. This is the total number of upward blocks divided by the number of built blocks. These two values are set in the data window of the current timeframe in the following format:

TF1 6 block = 50%; 10 block = 60%. With each new block, we need to update the data on the percentage of growing blocks on the total number of blocks.

**4. Block construction features**

The indicator is recalculated once at each new closed bar. Close(-1) shift 1 bar and all blocks and timeframes should be recalculated anew with new data starting from the new Close(-1). This is how it works if "**fix time" = no.** If "**fix time" = yes,** Close(-1) always remains on a single bar, the one it was located at during the first calculation or the one specified in **"bar date and time (-1)**. If **"bar date and time (-1)" is set to** zero, bar's Close(-1) is set to the one that was during the first indicator calculation. Close(-1) does not shift afterwards. With the closure of a new bar, Close(+1) appears. Previously built blocks remain in their places and blocks with BOpen(+1), BHigh(+1), Blow(+1) and BClose(+1) prices are constructed. The block construction direction changes. Now blocks are built to the right of the block closure price (-1). BOpen(+1)=BClose(-1) price, next BOpen(+2)=Bclose(+1). All blocks are built according to the algorithm similar to that described above. However, they are now built in the opposite direction. There is no limitation on the maximum number of blocks here. The maximum number of blocks is only needed for construction from the current moment to the past. Now blocks are built in unlimited quantities. The number of bars to the past from Close(-1) bar is equal to the number specified in "**Number of blocks to build".**

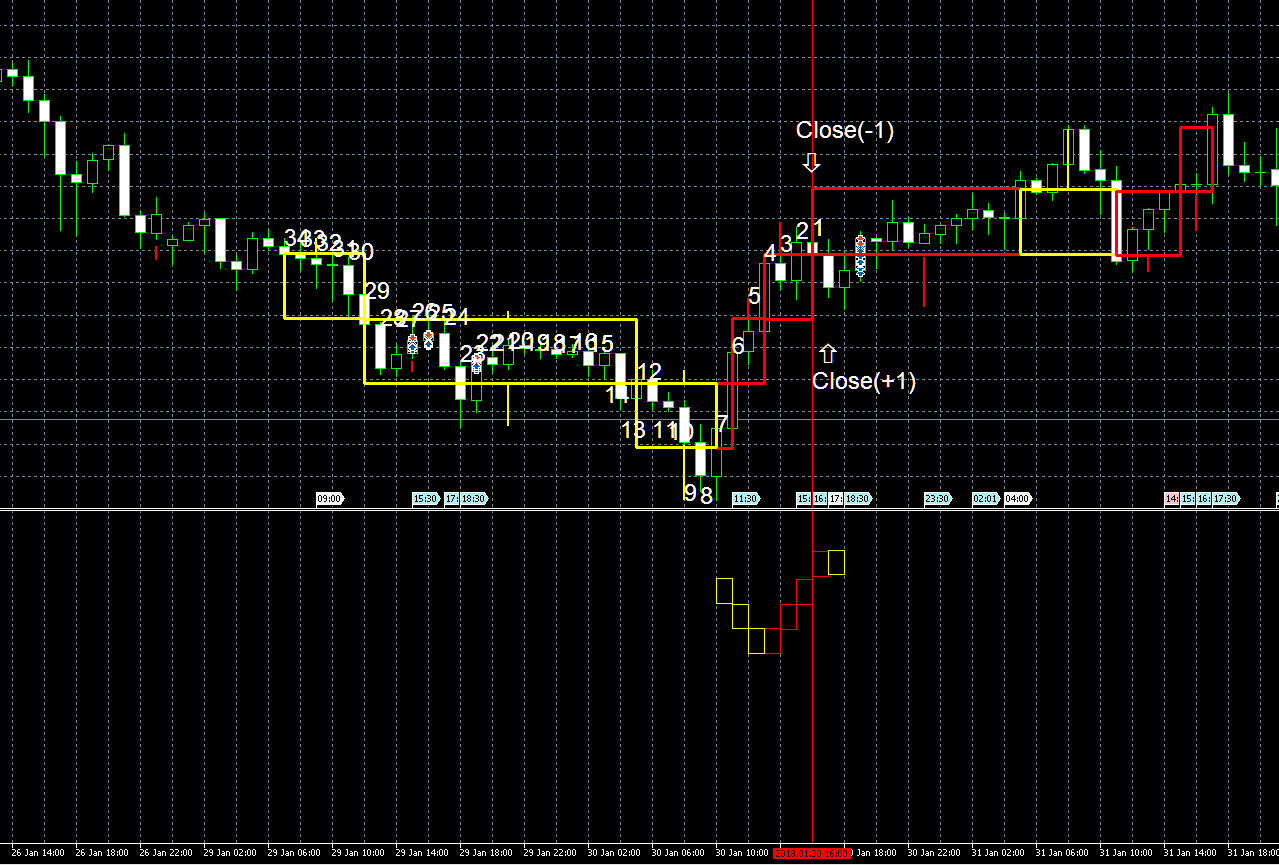


Figure 6. Building blocks if "fix time"=yes

Example:

Figure 6 displays an example, in which "**Number of blocks to build" = 6** and "**fix time"= yes**. In this case, we built 4 more blocks to the future relative to the original bar (-1). In this mode, there is no need to re-calculate all built blocks. New blocks should be added to save resources. The settings provide the ability to set "**bar data and time (-1)**" manually. The time is set up to minute. If 00.00.0000 00:00 is set everywhere, the setting is not used. The setting can be active only if "**fix time" = yes.** The parameter is used to set a starting point to build blocks both to the left and to the right so that any construction start point can be selected on history. This will be needed in the EA later.

**5. Block color**

"**Growing block line color"** and "**Falling block line color"** are defined in the settings separately. This is the color of the first timeframe. The remaining timeframes should be selected from the palette horizontally. For example, I set the red color in the upper left corner for growing bars. In this case, the entire palette is divided horizontally into the number of timeframes obtaining the number of shades to be applied for different timeframes. If there are 3 timeframes, the palette is divided into 3 parts and the first timeframe will be of the following color:

Shade: 0; contrast: 240; brightness: 120; red: 255; green: 0; blue: 0

Second timeframe: Shade: 120; contrast: 240; brightness: 120; red: 0; green: 255; blue: 255

Third timeframe: Shade: 239; contrast: 240; brightness: 120; red: 255; green: 0; blue: 6

Suppose that the initial point is set the following way for falling blocks:

Shade: 0; contrast: 142; brightness: 120; red: 203; green: 52; blue: 52

The remaining two colors are calculated similarly to growing bars along the palette horizontal.

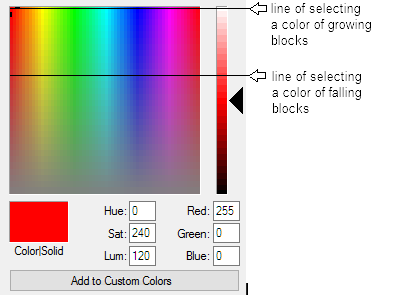


Figure 7. Selecting the colors of falling and growing blocks

**6. Features**

- if there is not enough history to build the number of blocks specified in the settings, then build as many as possible.

- no zero divide errors should occur during the operation.

The task has been developed and compiled by Maxim Romanov

e-mail: [223231@rambler.ru](mailto:223231@rambler.ru)

MQL5 profile: <https://www.mql5.com/en/users/223231>

LinkedIn profile: [www.linkedin.com/in/максим-романов-05475610b](http://www.linkedin.com/in/максим-романов-05475610b)