

EA Description and requirements

Description

This is a mean-reversion EA that seeks to trade as many retracements from a large price move as possible. We define a large price move in terms of the relative size of a bar on a chart on any timeframe (but lower than M15 is probably impractical because of spread and trading costs.)

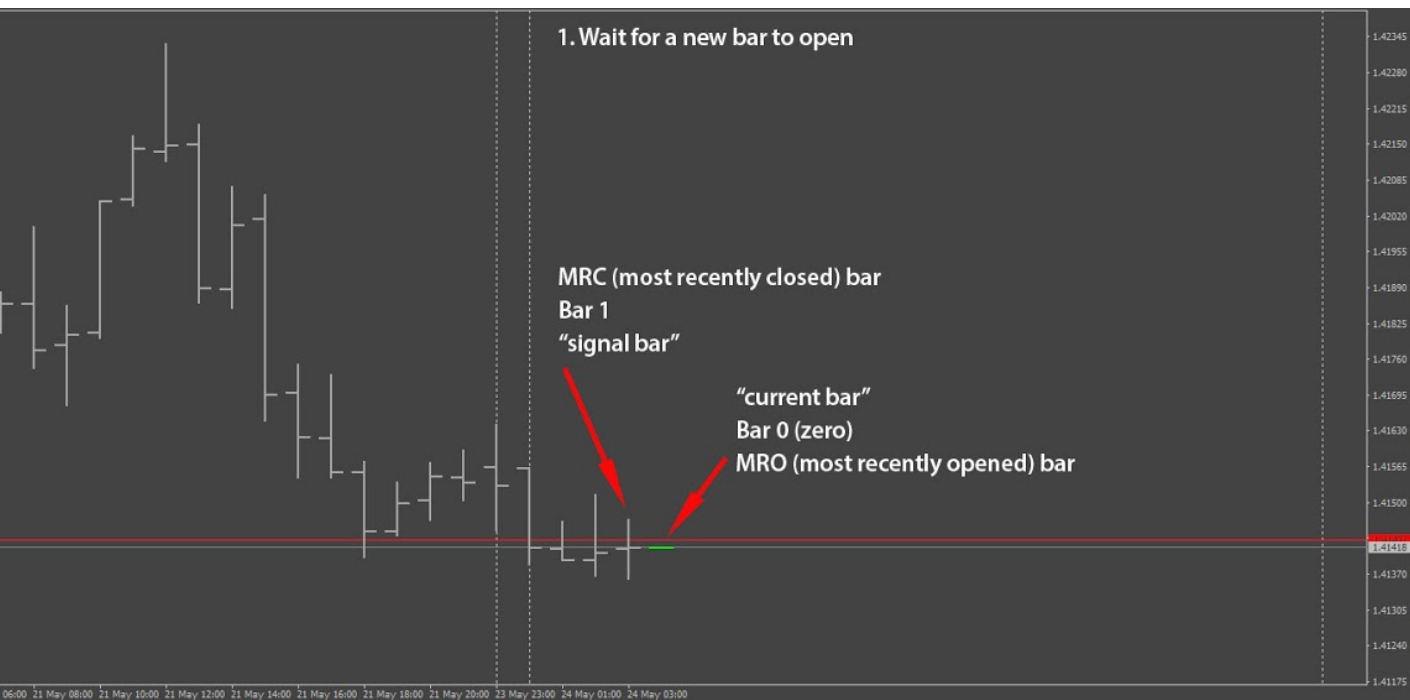
Custom terms are bolded. Input variables are in green and use camelCase. Inputs are preceded by the word 'input' but I may have missed some variables or inputs. Please don't rely on my weak MQL understanding.

Note: the numbering on the diagrams doesn't always correspond to the numbering of the text.

General requirements for placing orders

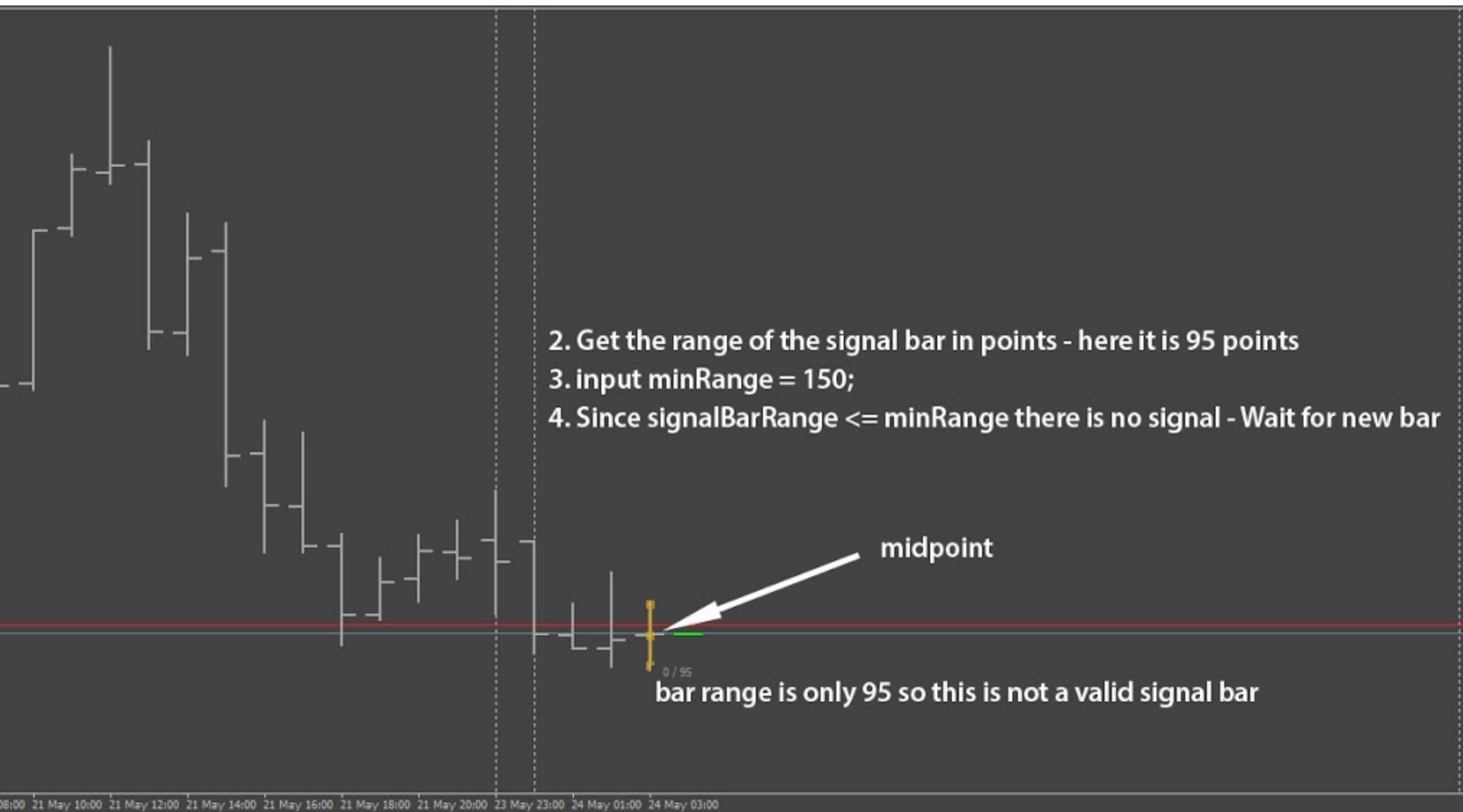
Pending orders: This system should not use pending orders sent to the server. Instead the EA should track the order levels and place market orders when those levels are reached.

1. Wait for a new bar to open



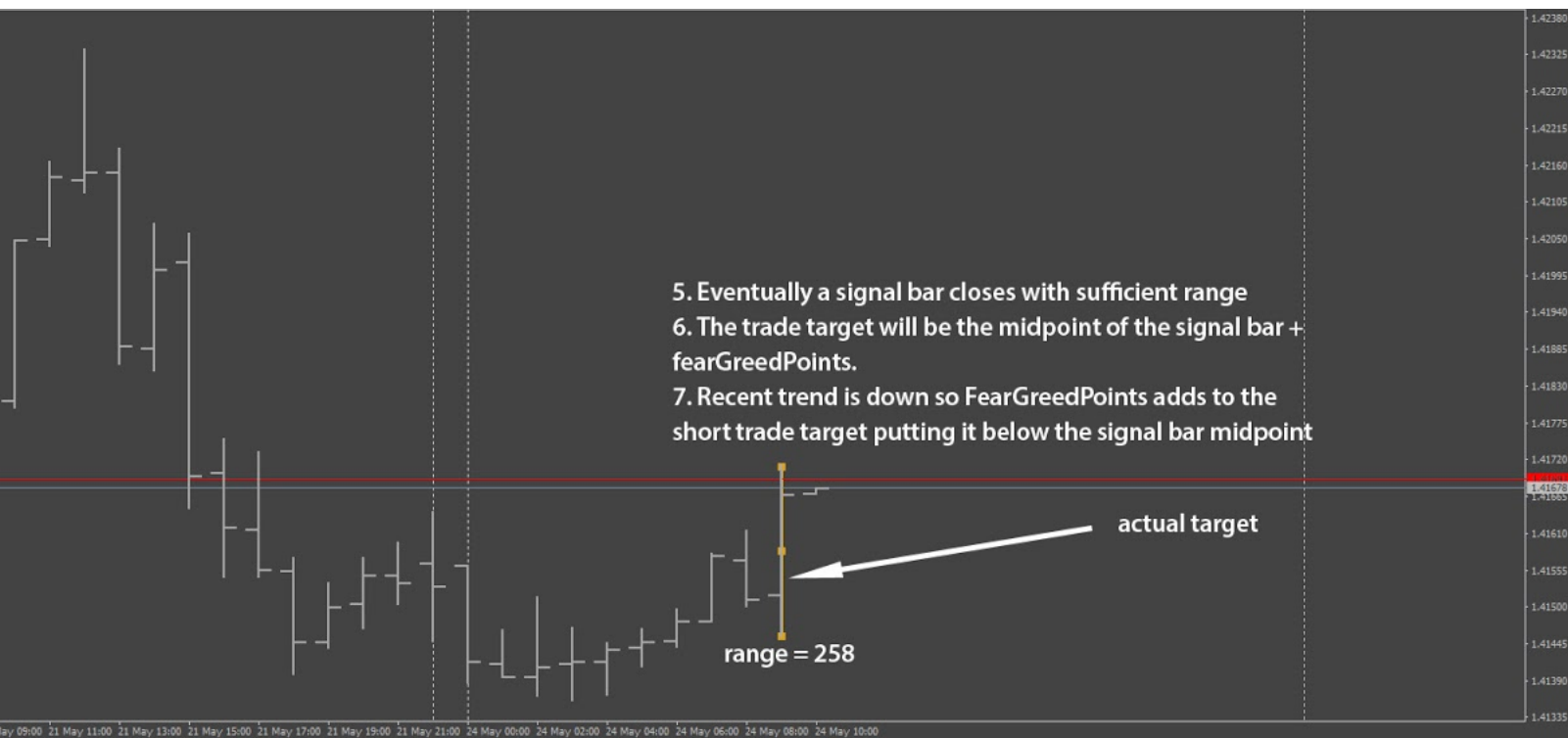
When there is a new bar, '**the current bar**' is **bar 0** or **MRO** (most recently opened)
The bar that just closed is **MRC** (most recently closed) or **bar 1** or the '**signal bar**'.

2. Get the range in points of the signal bar.
3. **input minRange** is the minimum height of a valid signal bar, in this case it is set to 150 points.
4. If the range is less than the **input minRange** value this bar is too small to trade. Wait for a new one.



5. If the range meets the minRange check this is a valid signal bar.
6. Get the **midpoint** of the signal bar. The midpoint is simply the halfway point of the range. If the range is an odd number of points, round up.
7. The midpoint is our trade **target**. We expect price to return to this point or near to it. We are going to adjust our target depending on the current trend.
8. Get the trend (up or down) of the most recent **input lookBackPeriod** bars. We are going to adjust the target up or down depending on whether we should be fearful or greedy based on the trend. This adjustment is set in **input fearGreedPoints** (explained below)

(document continues below)

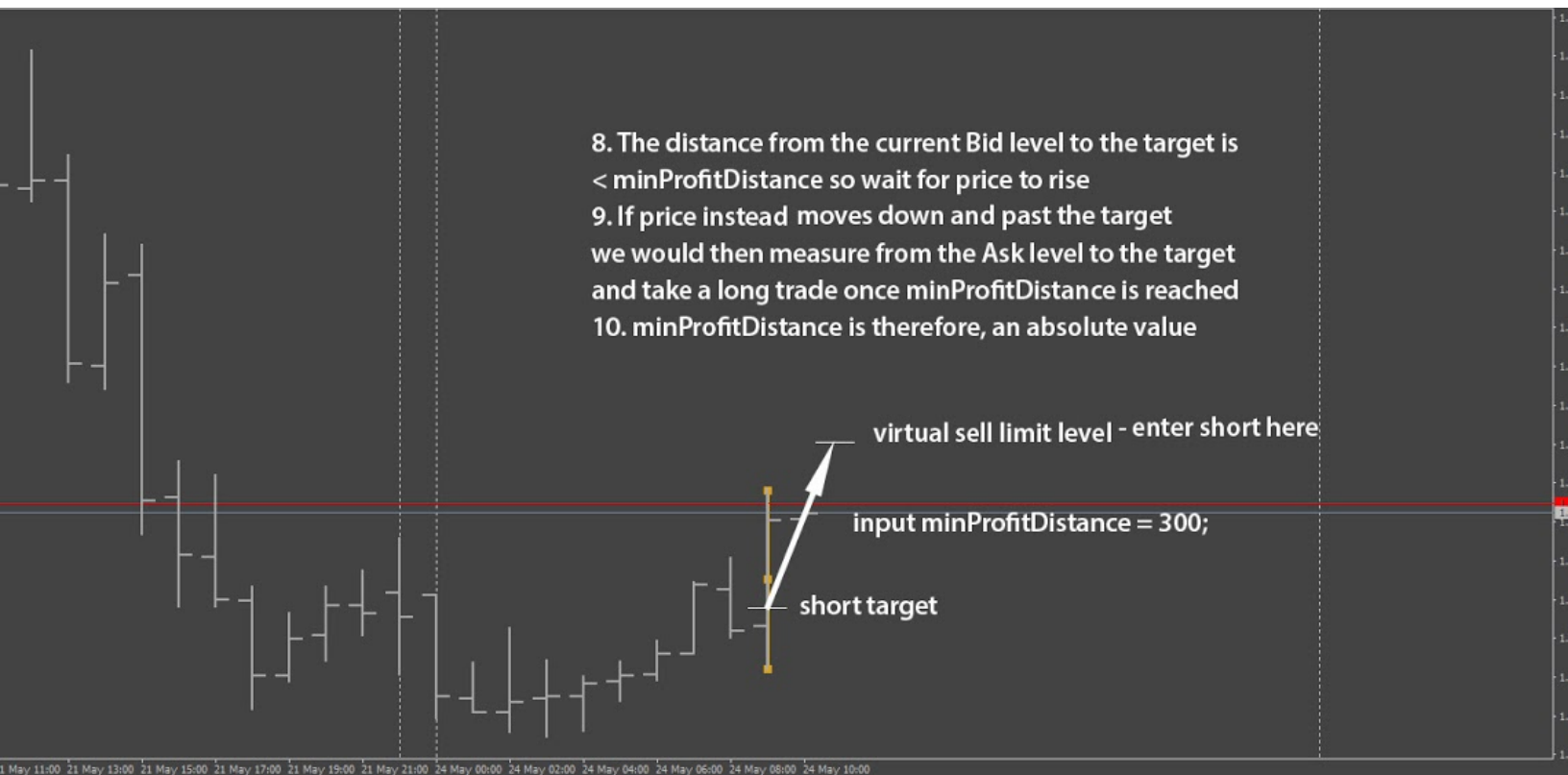


fearGreedPoints adjusts the midpoint up or down depending on the trend. Here the trend is down so we expect the target can be more ambitious. If the trend was up we would still aim for a short trade but we would move the target up from the midpoint. So **fearGreedPoints** can be positive or negative and it adjusts the midpoint value. For now let's say the default value for **fearGreedPoints** is 10 points. That means in an uptrend the target would be midpoint + 10 and in a downtrend it would be midpoint - 10.² This target adjustment is the same for both long and short trades.

9. Now we need to check if the current ask/bid is far enough away from the target. **Input minProfitDistance** can either be a multiple of the signal bar range or it is a fixed number of points. I'm not sure what is better. If I can have both inputs and disable one or the other that would be ideal.

When the bid or ask is too near the target (less than **minProfitDistance**) we must wait for price to move away from the target. When this happens an entry will be triggered as though it were a limit order. *Note that all trades and entries for this system should NOT use pending orders as I am paranoid about providing my broker with information.*

² Maybe **fearGreedPoints** needs to be two variables? Maybe it should be two inputs? It would be cool if it could be adjusted automatically depending on trend steepness, but that's probably more work than it's worth.



10. We expect price to visit the midpoint area. Whether that happens going up or down it doesn't matter! If price moves past the midpoint before triggering the minProfitDistance , but then achieves that distance on the other side of the midpoint, we maintain our current target but change the virtual limit order type. We were short in this example, but later this became a long trade.

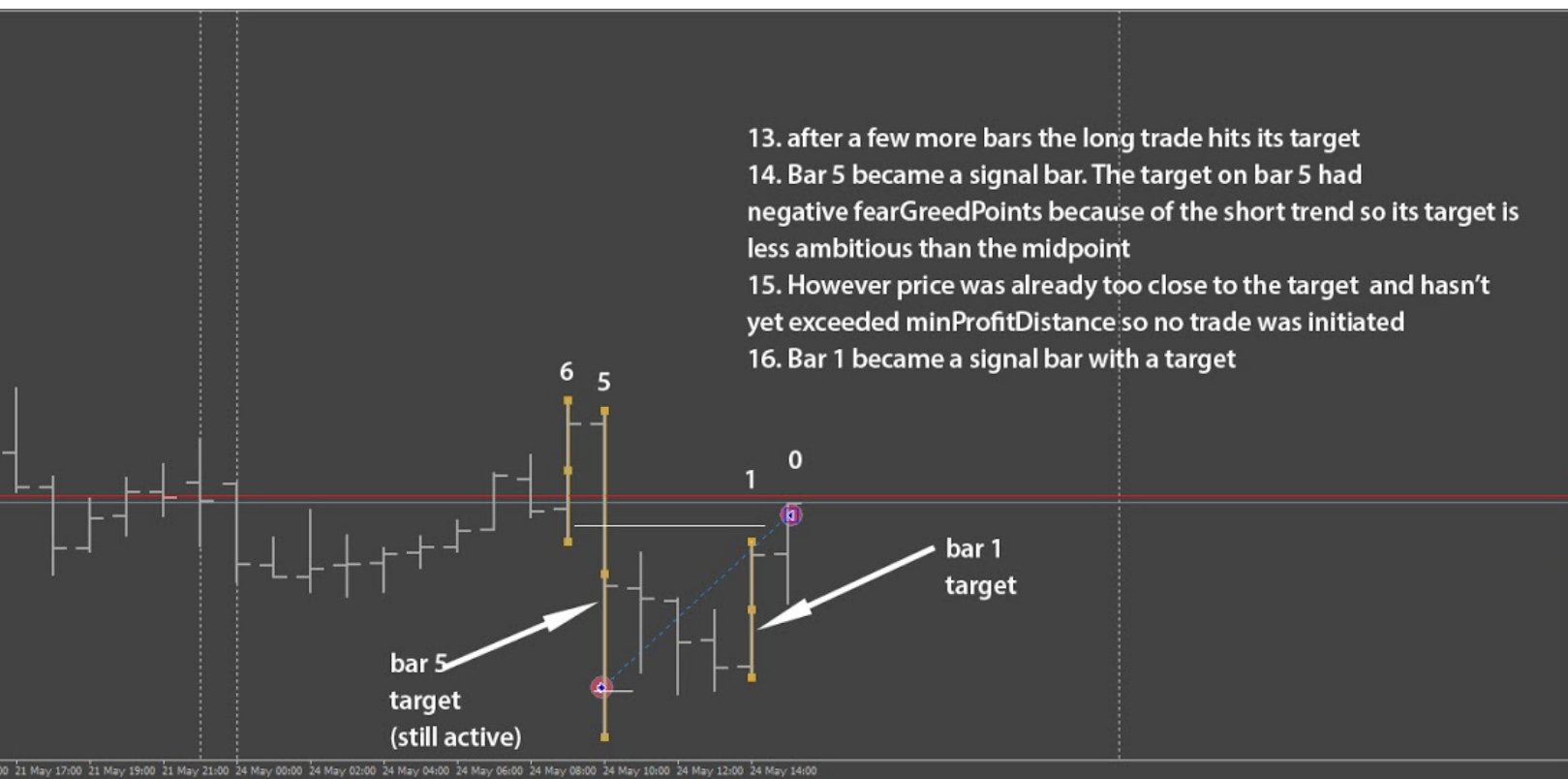
(document continues below)



11. Price moves down a bit further and **minProfitDistance** is reached. The trade (now long) is triggered with a target at the midpoint of the signal bar plus some extra points from **fearGreedPoints**.

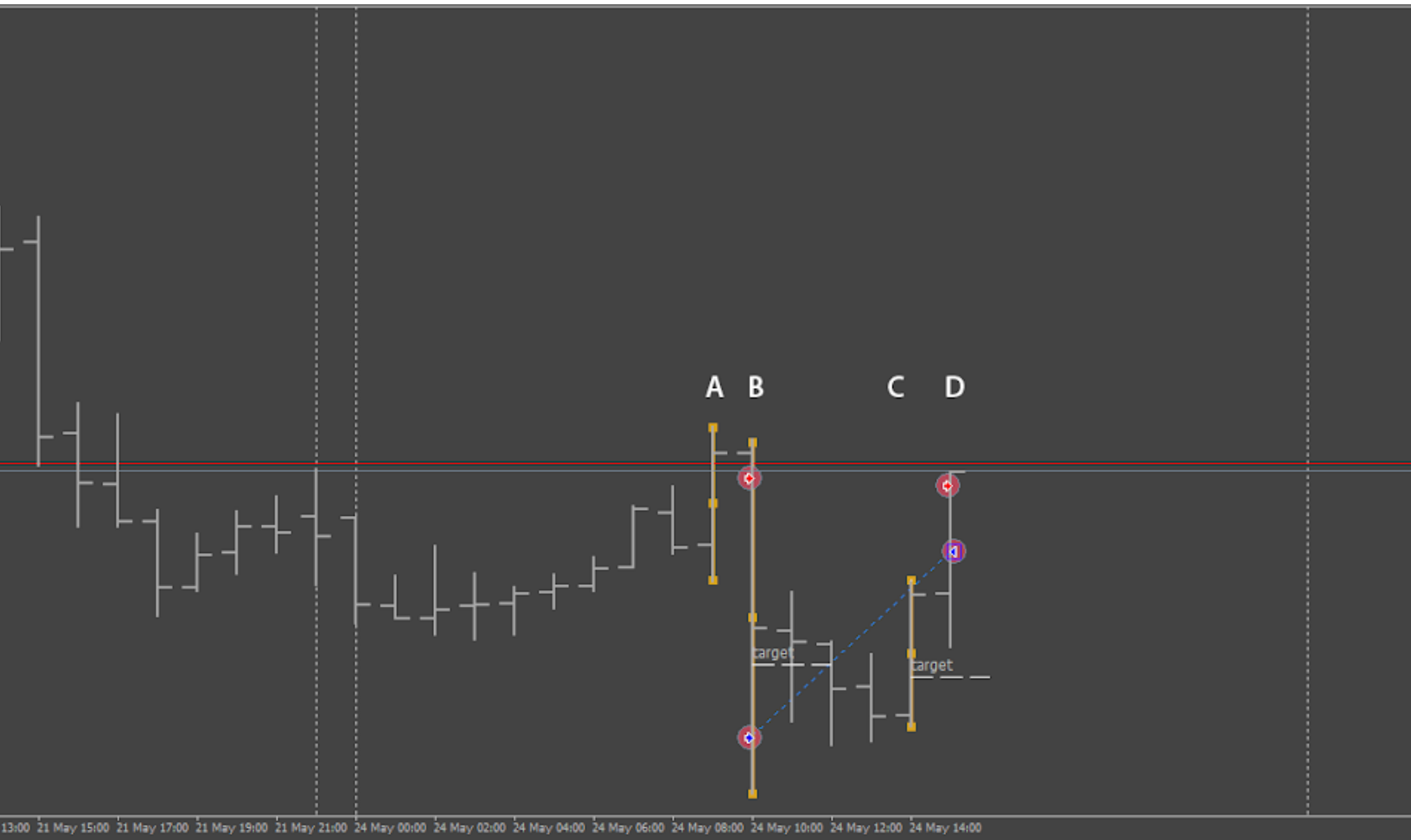


12. The signal bars are marked with a goldenrod trendline (set vertical) to show the range and midpoint.



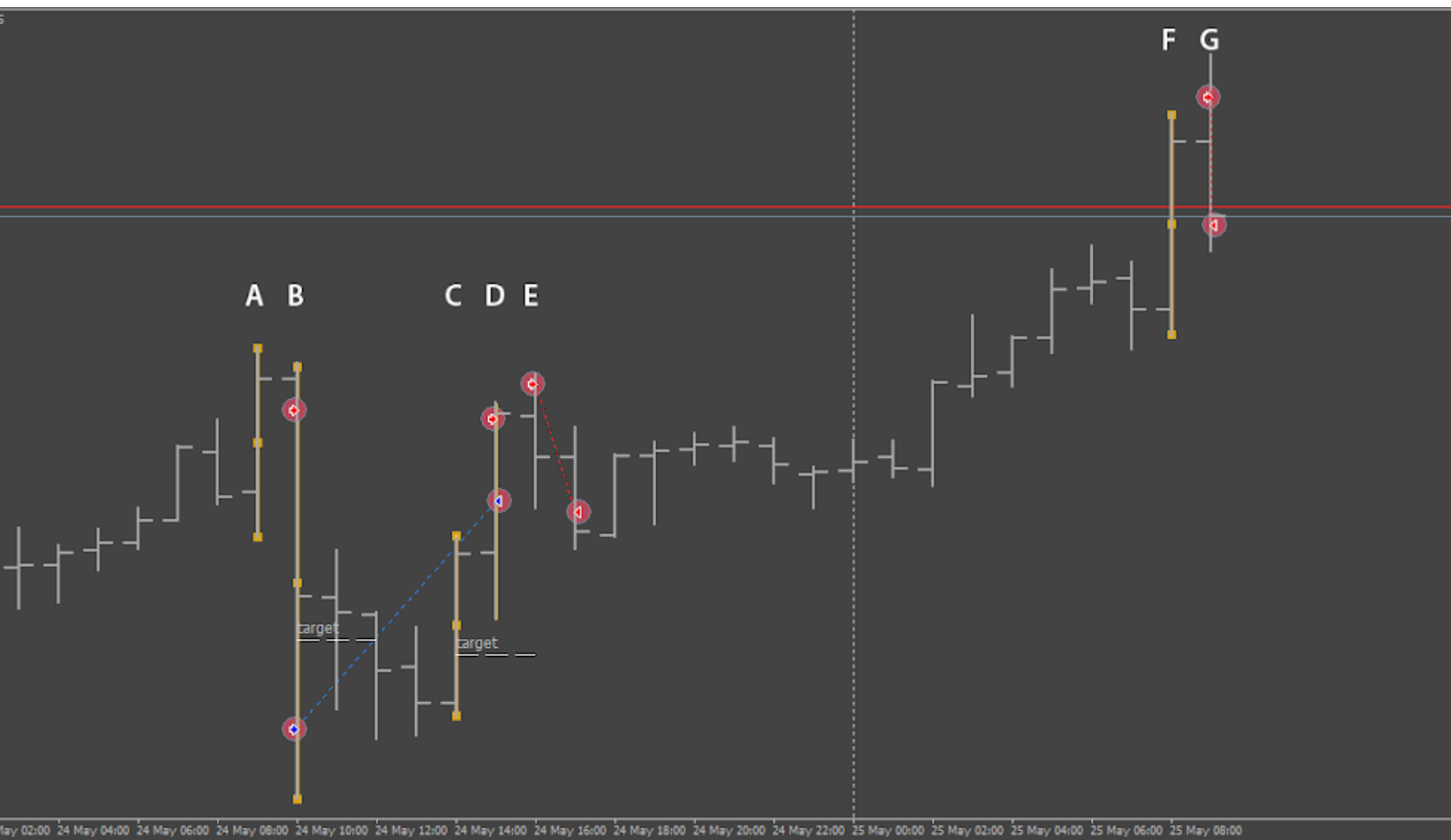
13. In the above diagram we can see that the long trade signalled by bar 6 is complete. The next bar, bar 5 is also a signal bar and has a target marked by an arrow. There is no long trade associated with this target (yet) as price hasn't moved far enough (past `minProfitDistance`).
14. Bar 1 is a new signal bar. It has a target but price must move up to meet `minProfitDistance` before a short trade will trigger. You will note that this situation creates two possible short trades with similar targets. We need an option to suppress these kinds of repeated opportunities to avoid taking on too much risk at similar levels. `Input maxTradesPerLevel`. Let's set that to 2 for now. If any new targets occur within a distance `Input SeparatorDistance` of these targets the oldest target (on bar5) will be removed and the signal will be discarded.

(document continues below)



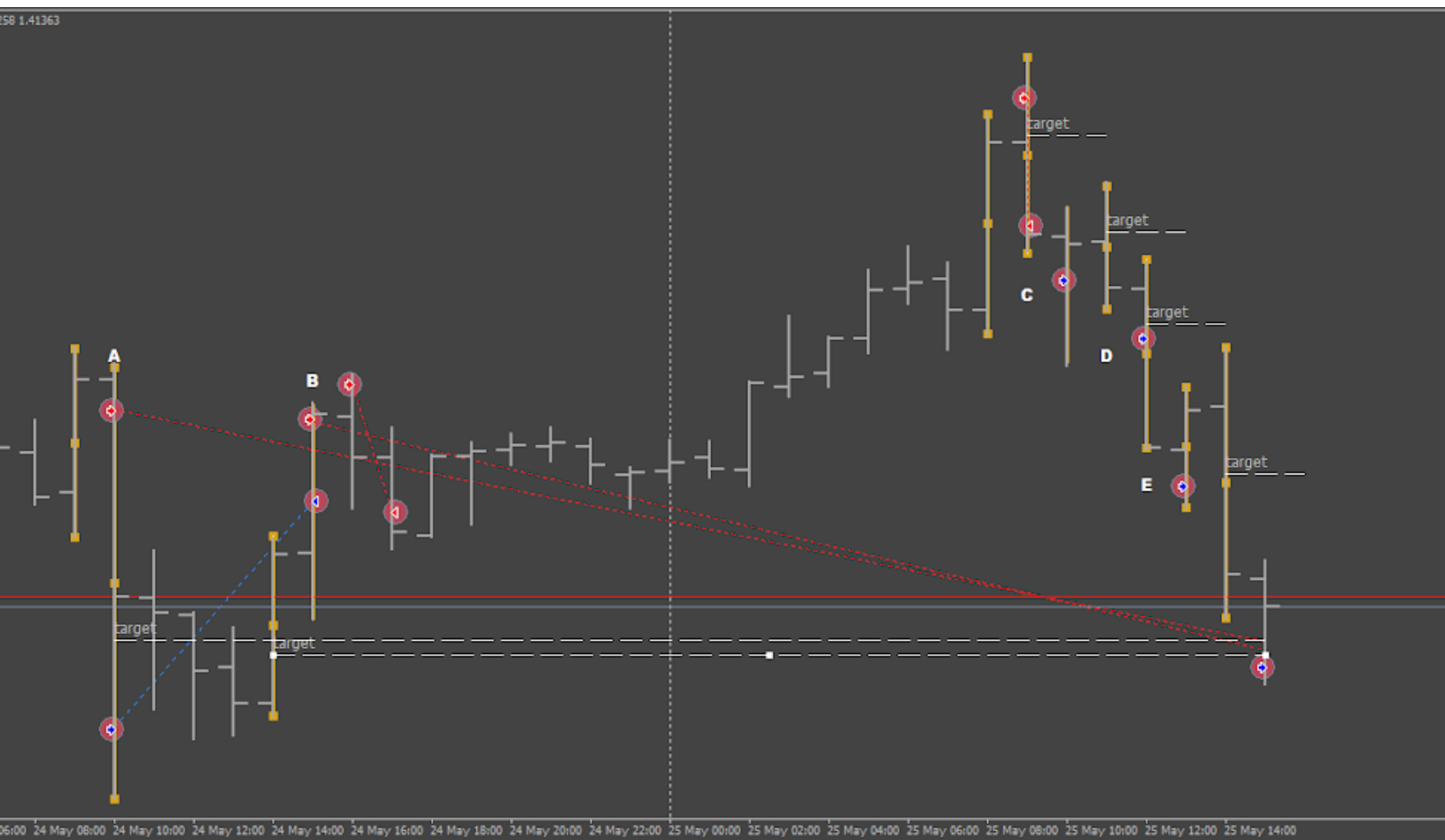
15. I've re-labelled the bars with letters that won't change as time moves. Price has moved up and met the `minProfitDistance` two short trades. Note the active short trades on bar B and bar D (red arrows) and their associated target levels (white dashed lines).

(document continues below)



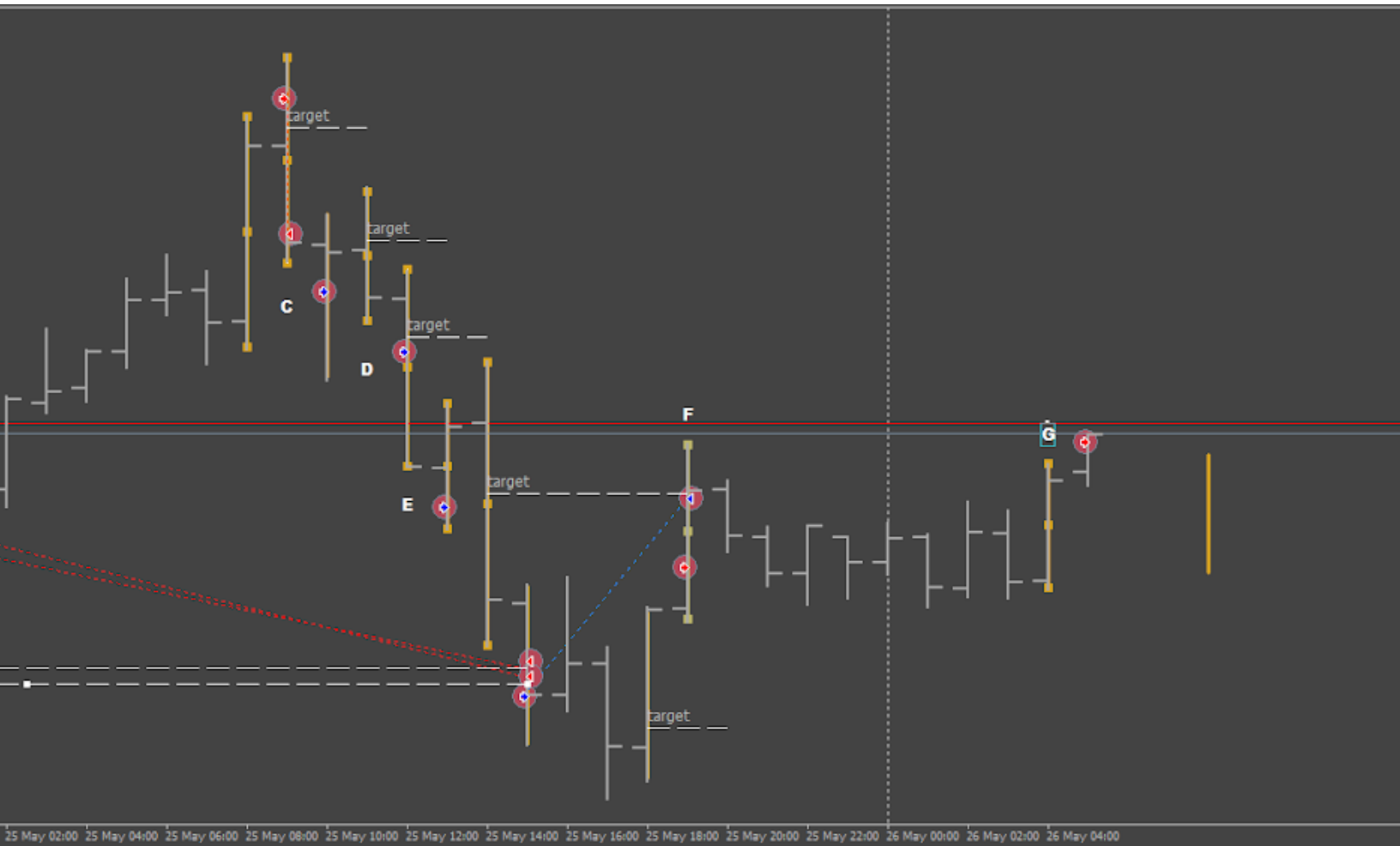
16. In this screenshot Bar D becomes a new signal bar. Price rises up to reach `minProfitDistance` triggering a short trade on bar E. That reaches its target at the bar after E.
17. Bar F becomes a new signal bar and when price rises sufficiently it triggers a new short trade on bar G that reaches its target before that bar closes. The trades on bars B and D are still active but losing.

(document continues below)



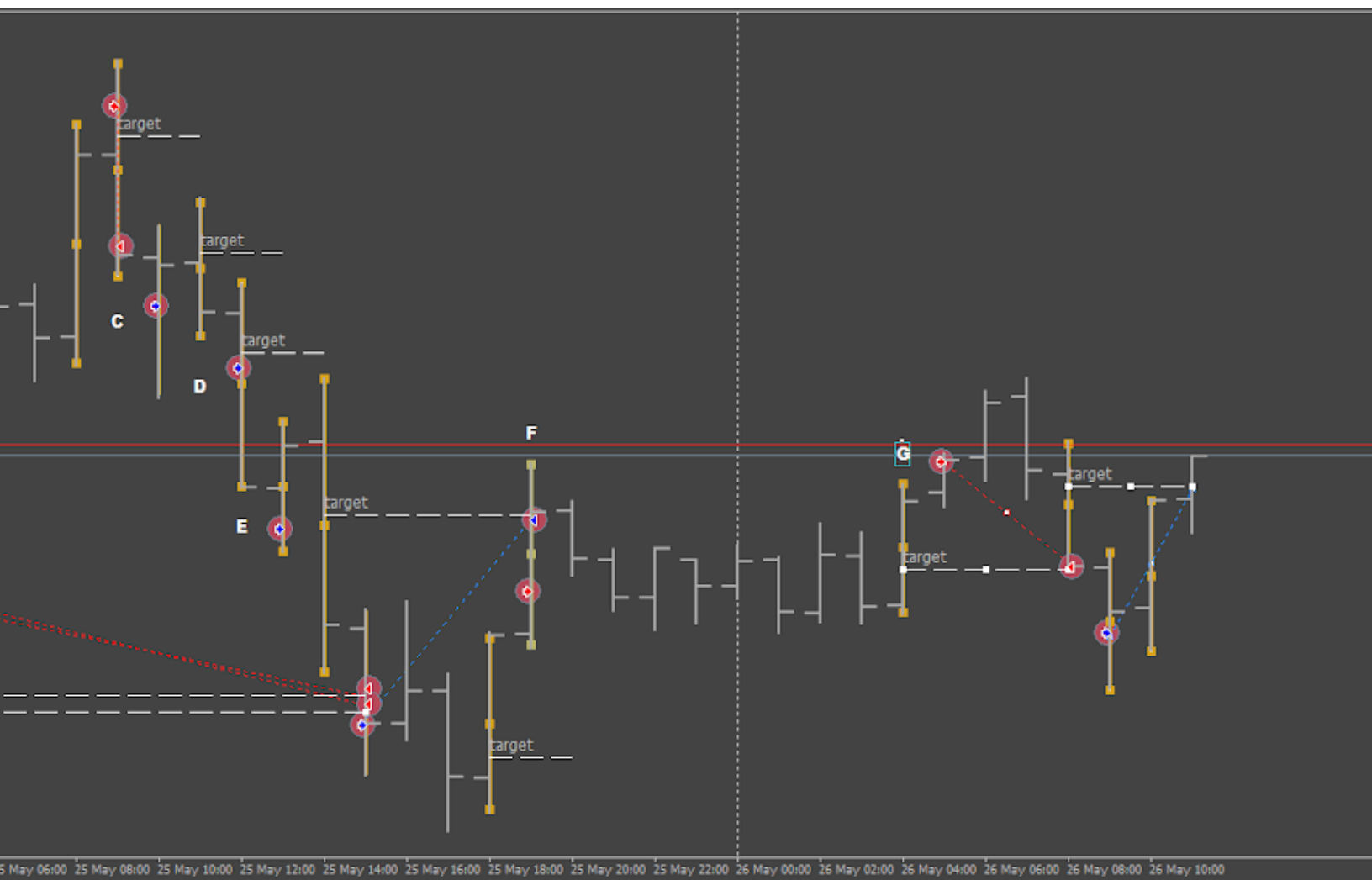
18. New long trades and targets are opened on bars C, D, and E as price moves down. A winning short trade occurs on Bar C.

(document continues below)



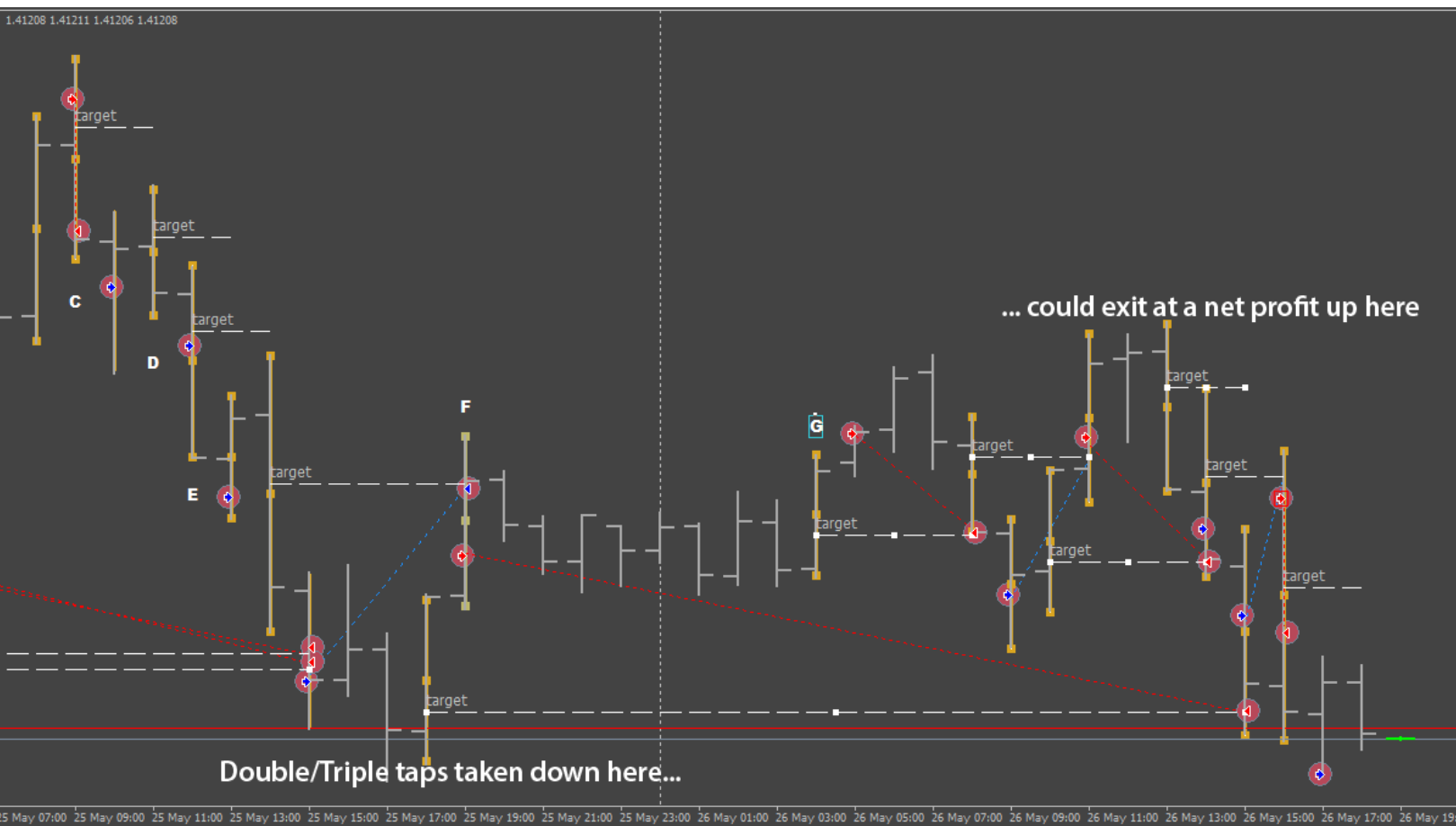
19. Here we can see that the signal produced by bar F is going to have a similar target to bar G. If `maxTradesPerLevel` was 1 then G would supplant F and the signal at bar F would be discarded.

(document continues below)



20. The signal at bar G completes its short trade and a long trade afterward reaches its target.

(document continues below)



21. Now the long trades at C, D and E are becoming a problem. This is why we have a range of stop-loss options.
22. To discard losing trades we can establish that a trending condition has begun and enter a trend trade. This would be a trade equal or greater to the volume of the losing trade that spawned it. `Input HedgingTrendTrade: true` by default. This trade has no SL or TP, but it closes when either it is more profitable than the losing trade that spawned it or when its parent trade closes. This trade activates when a parent trade hits the `hedgingTrendDistance`.
23. Entries
So far we have been looking at 'single tap' trades. That is, one entry for one signal. We should have an option for 'double and triple tap' or even more. What that means is if price moves against our target, we enter again at a new price, and if it moves against us again, at yet another price.³
24. Exits
These additional 'tap' trades, when they become profitable enough, should close out any losing trades of the opposite type.

³ Should these add-on 'tap' trades have a martingale option so that their volume increases each time?

You can see that if we had enough additional long trades when price nearly reached the target at bar 'E' then we could have closed all those longs at a net profit.

25. Stop Loss Options

2 types of stop loss - disable any one by setting it to zero.

- a. **Input SignalBarMultSL** (virtual SL is set at a multiple of the signal bar range in points)
- b. **Input PctSL** (% of acct balance - default 2%; Close a trade when it exceeds 2% of acct.)

26. Hedging option

- a. **Input HedgingTrendTrade**: default: true If price moves a certain distance from our target after a trade is entered, then place a 'mirror' opposite trade type at a volume multiplier. Close this trade when its profits exceed the losses of the losing trade.
- b. **Input HedgingTrendVolume**: default 2x the losing trade
- c. **HedgingTrendBuffer** - add this to the profit of the parent trade + hedgingTrade to determine when both trades can close to break even. Recall that hedgingTrades don't have a target. Their goal is to close the losing parent trade without suffering a net loss.

27. Double/Triple taps (add-on trades)

- a. **Input numberOfTaps** (default: 2) zero disables
- b. **Input separatorDistance** (points) default: 50 The distance between the follow-up taps.
- c. **Input tapVolume** The volume of add-on trades. A multiple of the original trade that spawns the double tap.
- d. **Input maxTotalTaps**: 10 controls the total number of follow-up trades allowed

28. **Input MaxTotalTrades**: 20 total trades of any type, original, tap, and hedging trend

29. **Input Failsafe** if account drawdown exceeds this amount, close all trades, exit EA with message

30. **Input Breakeven**: default false; when true if the trade is in profit by trigger(points) place a SL at breakEvenPoints

- a. **Input BreakEvenTrigger**: 40 points
- b. **Input BreakEvenDistance**: 25 points

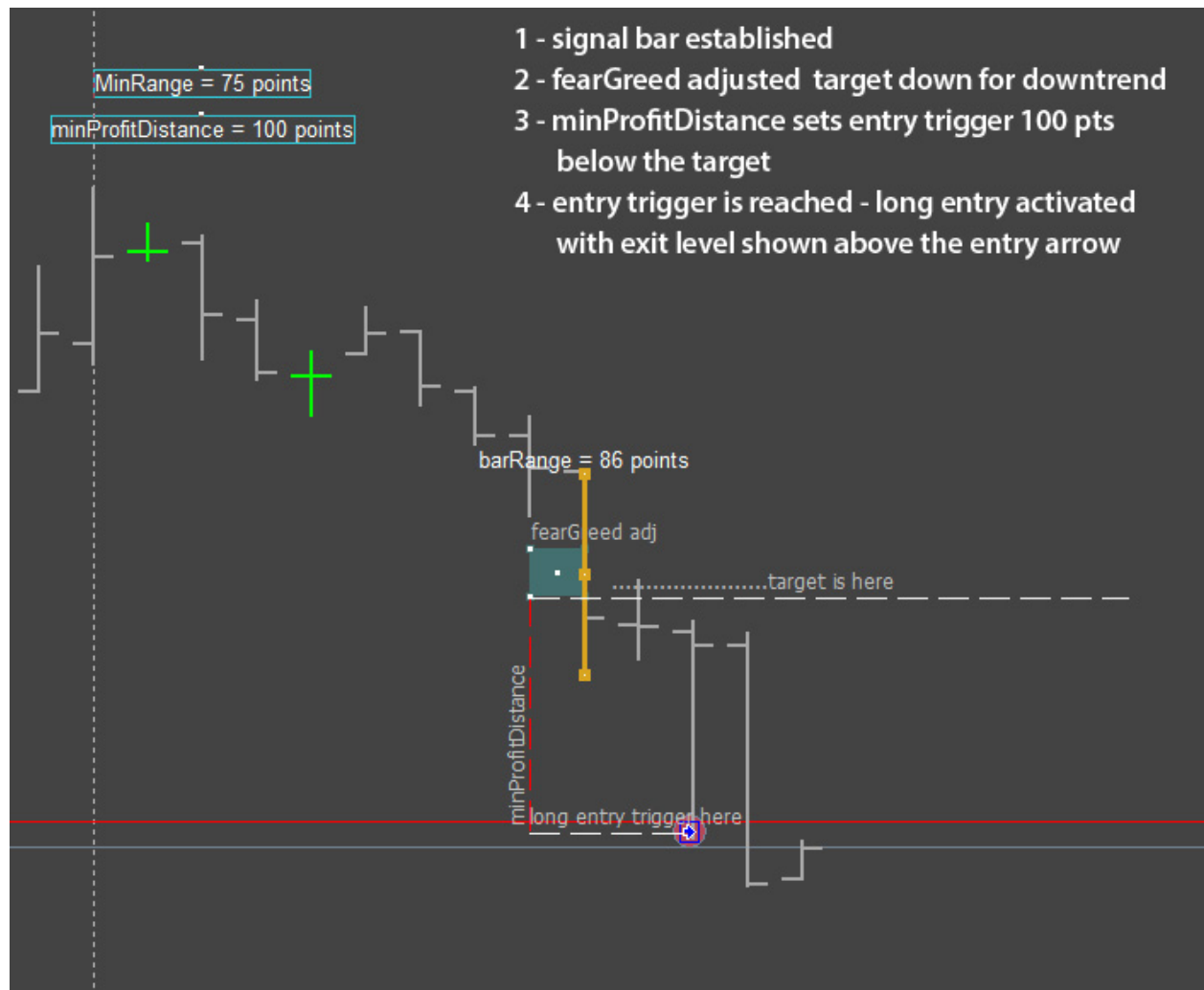
31. Hours

- a. **Input tradingHours** - no need to specify the day, but avoid trading on Sundays.
- b. Hours should be in 24h time with minutes. Allow for AM/PM sessions
- c. Default broker time offset GMT - 2

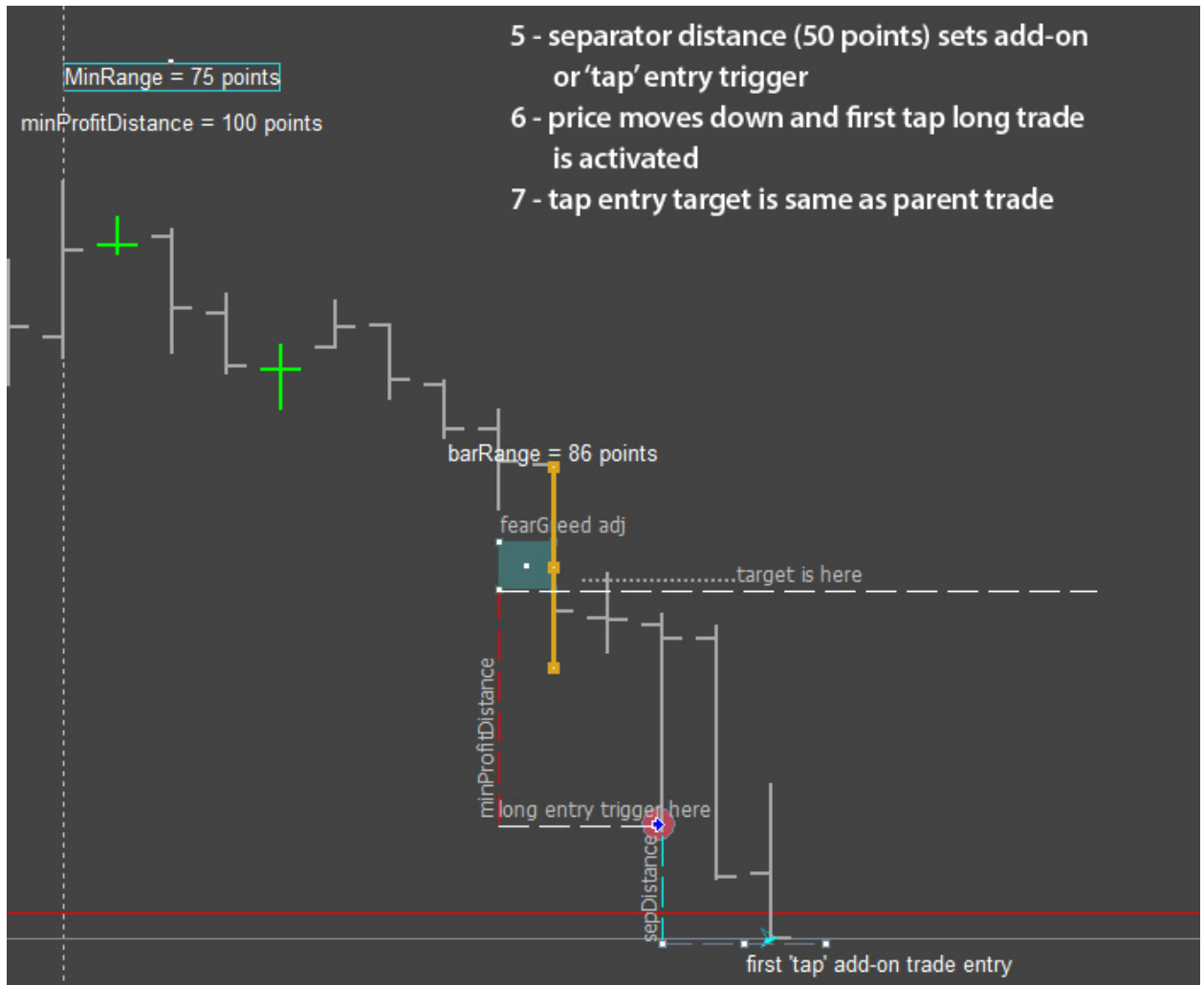
(document continues below)

More Trade Examples

Including 'tap' (add-on) trade example and hedgingTrend trade example

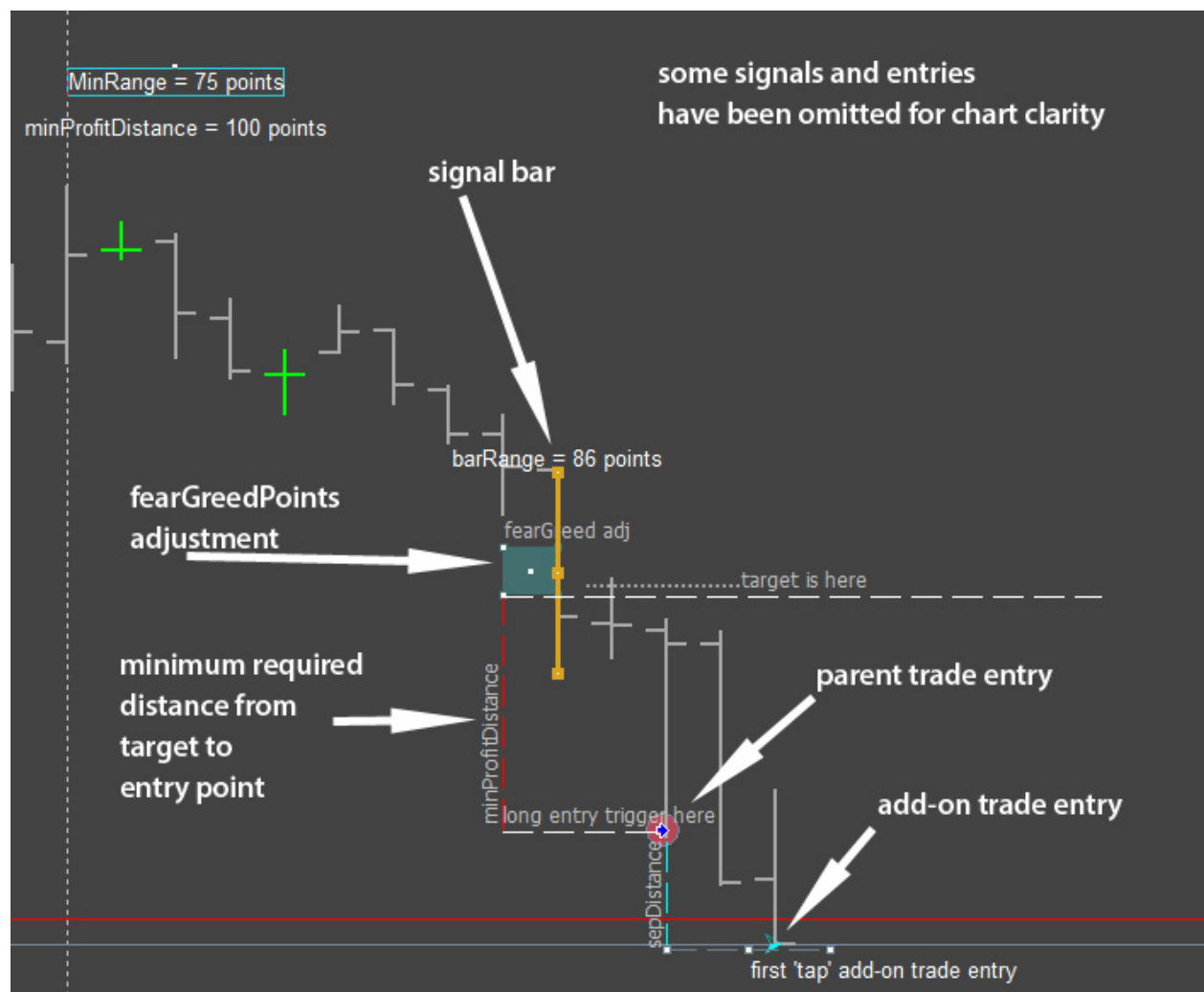


1. A new signal bar for the period is signalled when it exceeds **minRange**. In this case a bar of 86 points is > than **minRange** which is 75 points
2. **fearGreedPoints** is represented by a green rectangle of 20 points, but in the EA it will not be visible. The midpoint of the rectangle is centered on the midpoint of the signal bar. In this downtrend we will aim for the bottom of the rectangle (an adjustment of -10 points). If this were an uptrend we would aim for the top of the rectangle.
3. **minProfitDistance** is represented by a red dashed line but in the EA it will not be visible. At the bottom of the minProfitDistance line the long entry trigger is displayed with a white dashed line. If this is displayed in the EA that's OK, but it's not required.
4. We can see the long entry arrow where price touches the entry trigger level. Above this, a target 10 points below the signal bar midpoint is shown with a white dashed line.

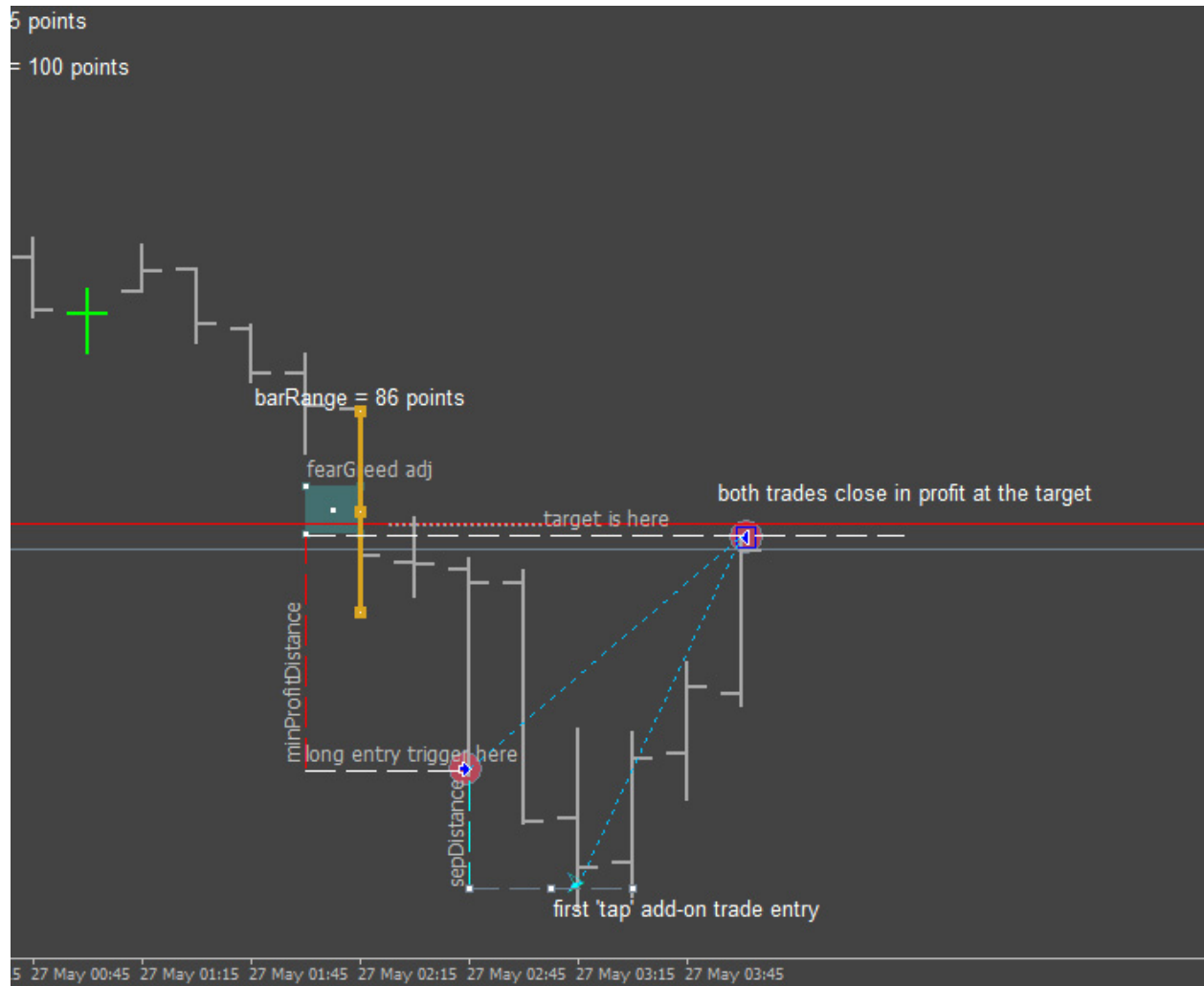


5. Below the parent trade entry the distance that price must travel to trigger a 'double tap' add-on additional trade entry is shown with a blue dashed line. This will not be visible in the EA.
6. The trigger for the add-on trade is shown with a grey dashed line and price reaches it triggering the double-tap. This is shown with a blue chevron arrow. The add-on and parent trade and hedging trade arrows should all be different types using red for short and blue or green for long. They don't need inputs as I can edit the MQL to get the arrow code I want.
7. The target for both of these active trades is the same level as the one established with the parent trade.

(document continues below)

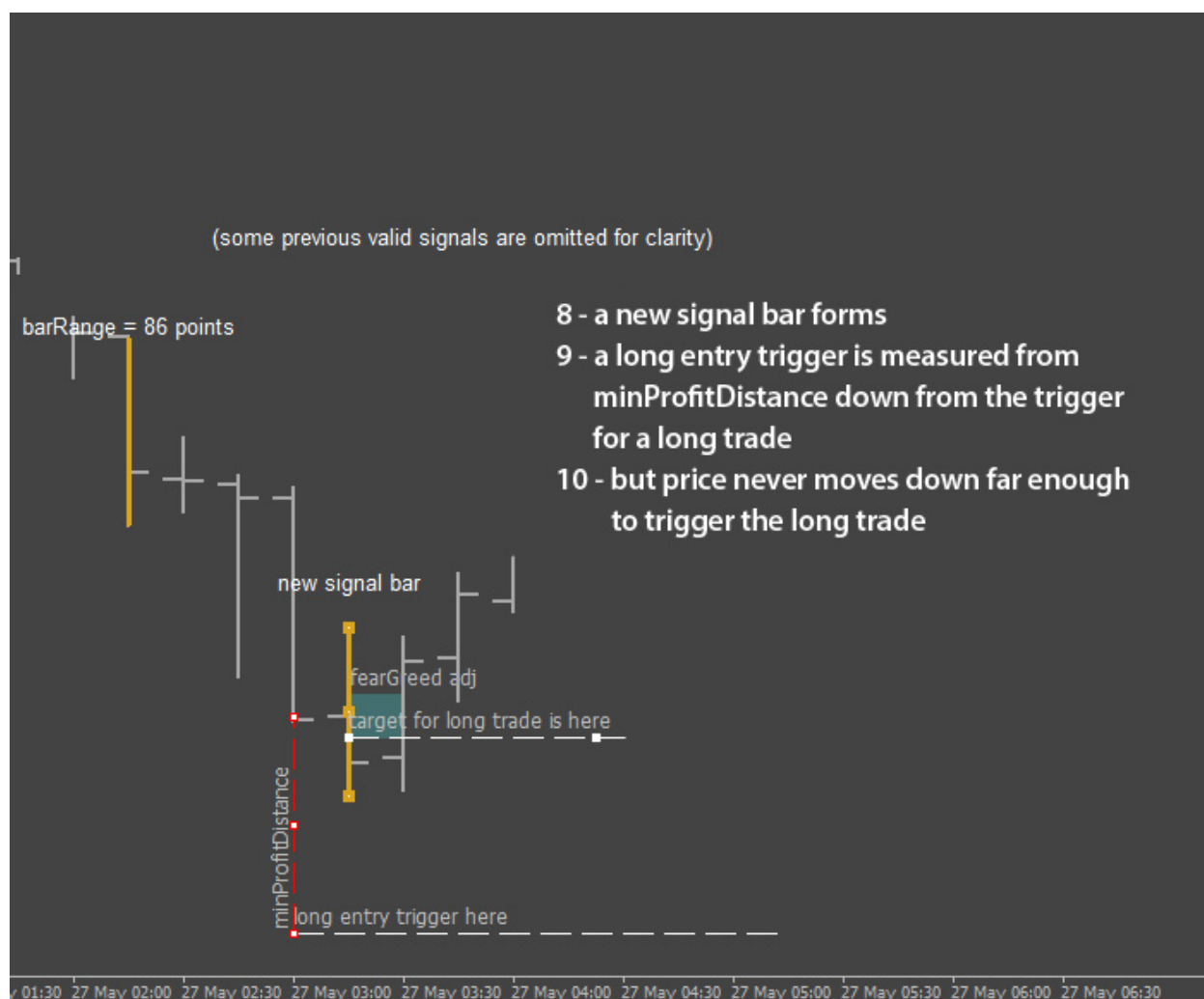


(document continues below)



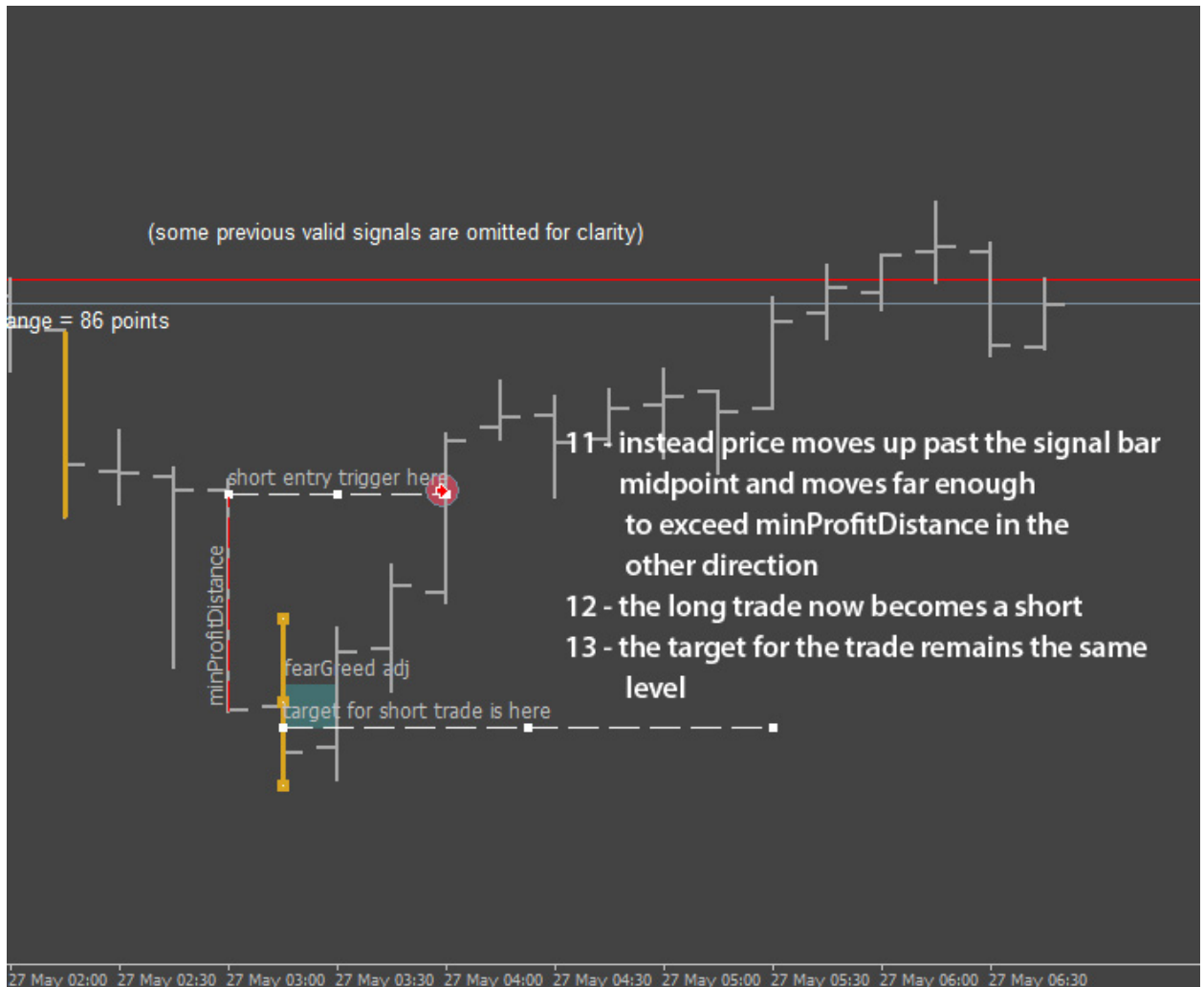
Price rises and both long trades close at a profit when they reach the target level

(document continues below)



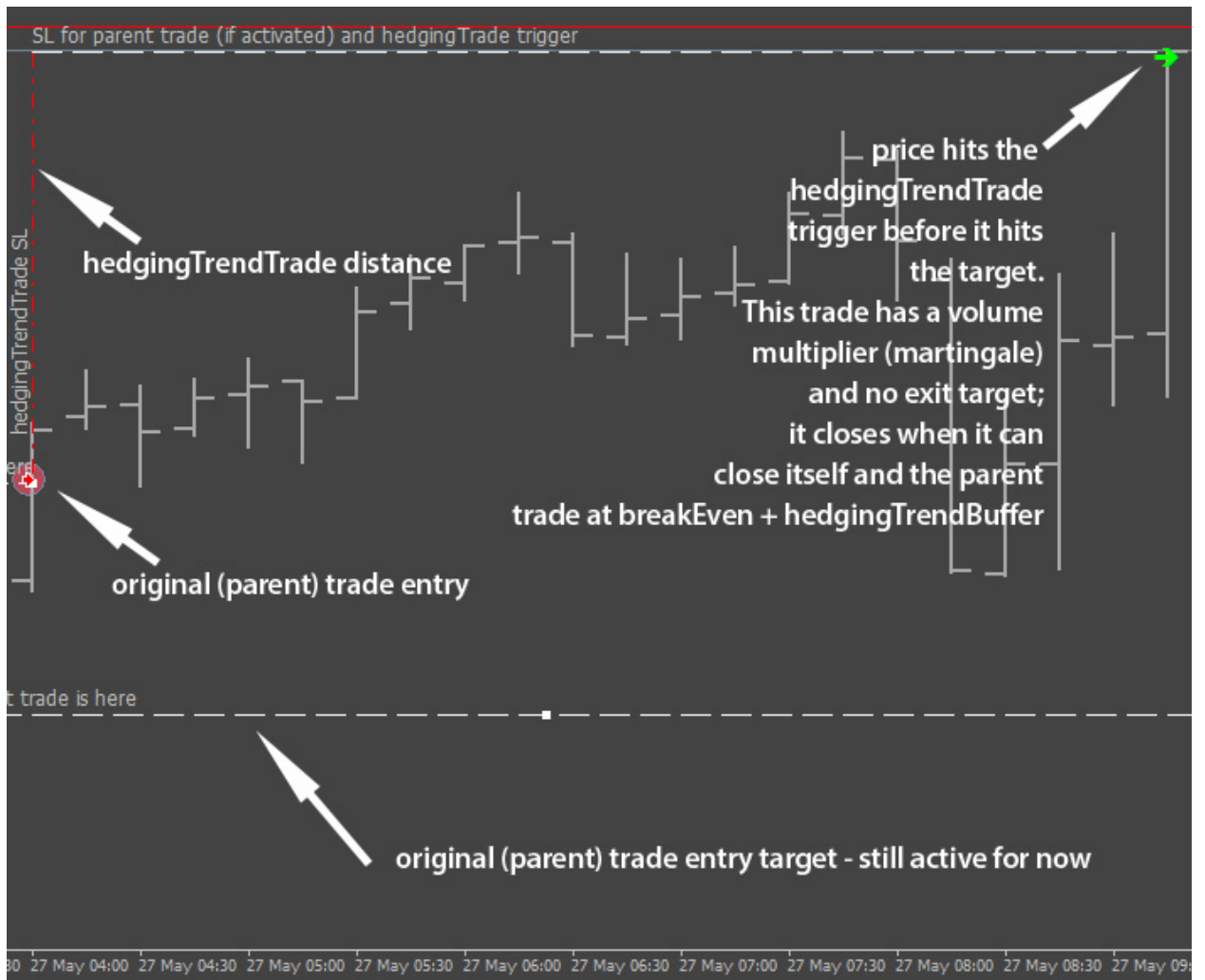
8. We have a new signal bar. Note that these examples omit several valid entries in order to avoid cluttering the screen with objects.
9. The long entry trigger is set **minProfitDistance** away from the target. It looks like I didn't draw it in the exact correct spot here as the distance should start from the target line and extend down.
10. In this case price doesn't reach the short target before continuing up. We do not discard this target level yet though.

(document continues below)



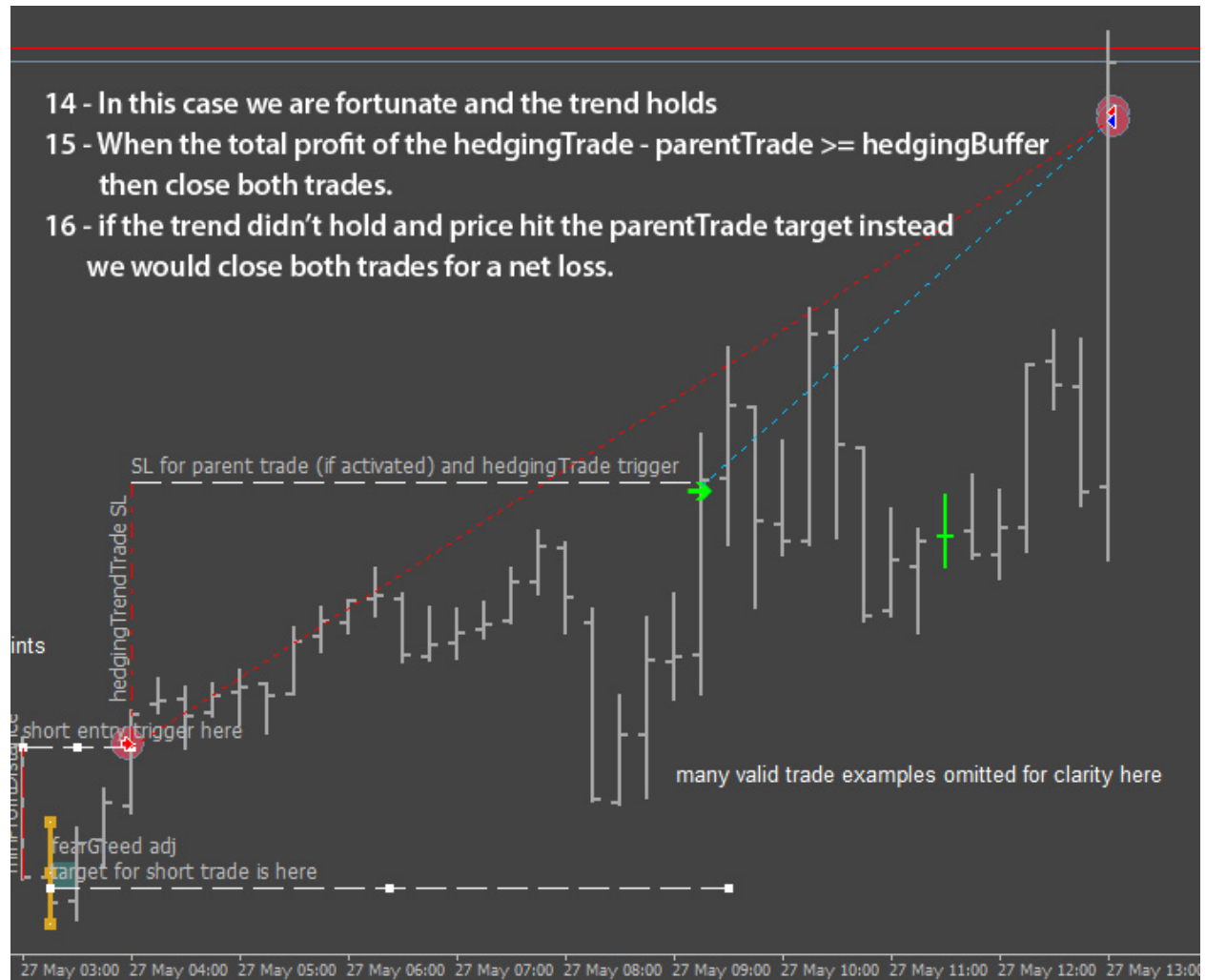
11. Prices moves up `minProfitDistance` going the other way so we discard our long trigger and set up a short trigger but continue to use the same target level.
12. The long trade trigger is removed and replaced with a short trade trigger which is activated. The short trade arrow entry point is marked on the chart
13. The target level is re-used. Unfortunately it looks like price is beginning to trend long.

(document continues below)



Indeed price does trend up 200 more points. At this level we consider the short reversion trade to be a bad risk and when price hits the **hedgingTrendTrade** distance (a bit like a SL level, but no exits are taken) we activate a long trade that has the option of being a martingale trade. In this case we assume the hedgingTrend trade volume is double the parent trade volume.

(document continues below)



14. The long trend threatens to retrace (and these bars would activate their own signals and entries but they have been omitted to avoid confusion in this example) but in the end the trend holds
15. and the **hedgingTrendTrade** closes at a profit along with the **parentTrade** that closes at a loss. Even with trading costs this trend trade will still be net profitable as it was double the volume of the parent. **Note:** the formula in my screenshot is wrong. It should be something like $\text{hedgingTradeProfit} + \text{parentTradeProfit} + \text{tapTradeProfits} \geq \text{hedgingTrendBuffer}$ but I'm sure you knew that.⁴
16. If the trend had failed and price did hit the original parentTrade target, we would have closed both trades for a net loss.

(document continues below)

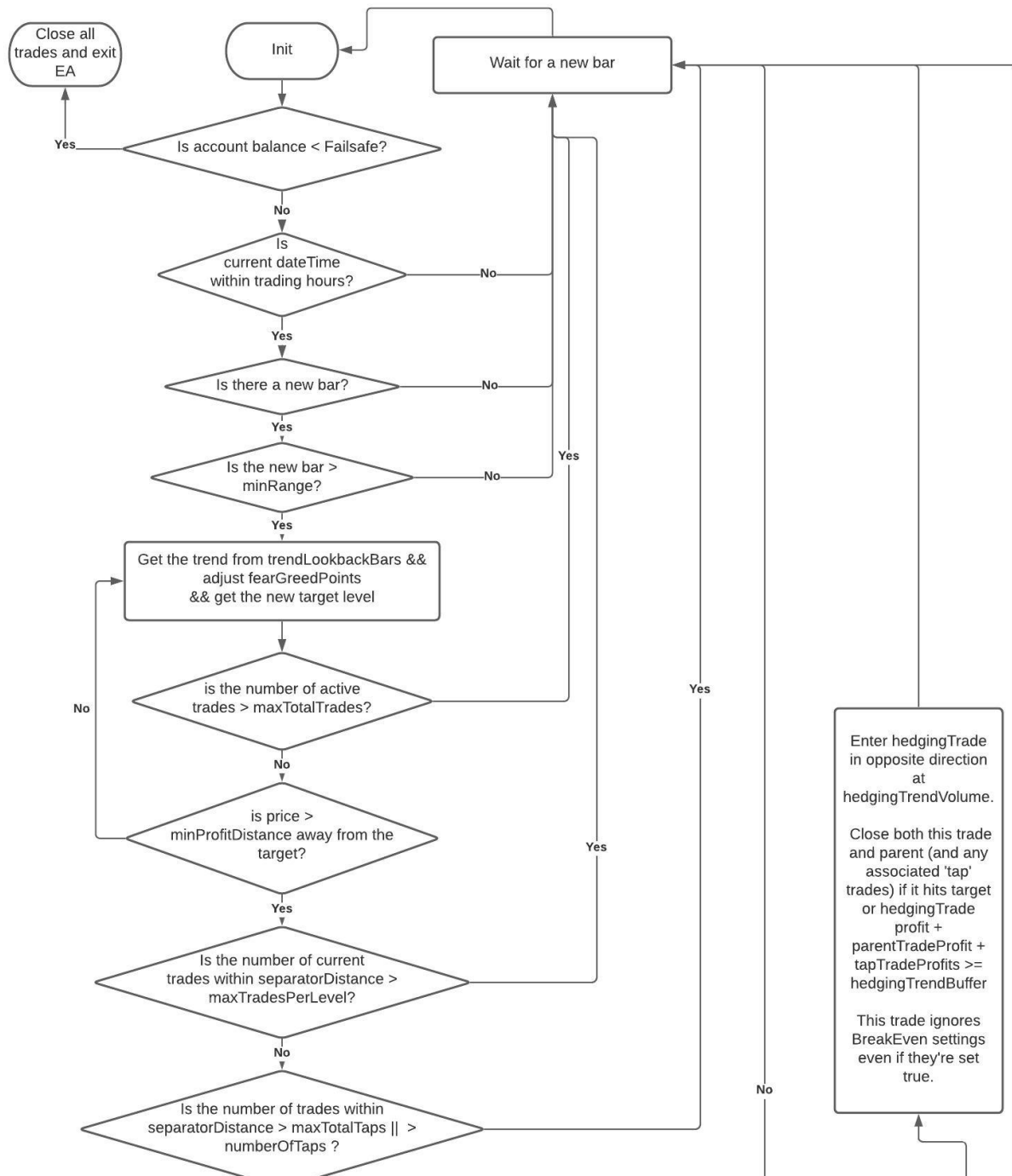
⁴ And have a much better idea of what formula to actually use.

List of Inputs and their default values

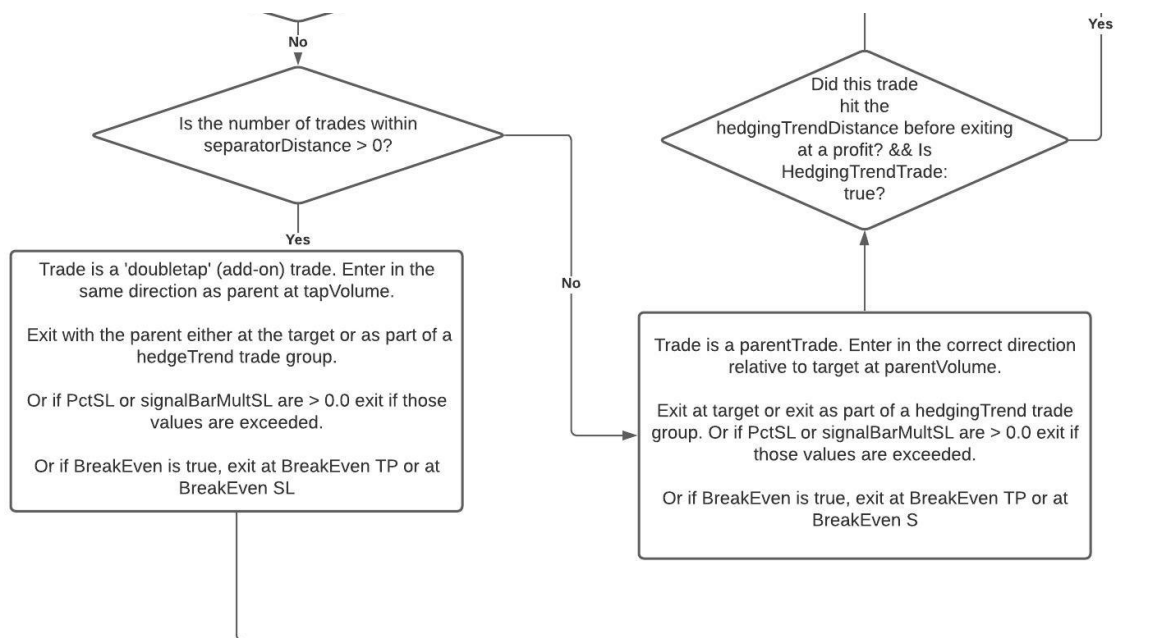
1. minRange: 150 (points) //The minimum size of the tall bar
2. lookBackPeriod: 60 (bars) //The period to check for a trend
3. fearGreedPoints: 10 (points) //The adjustment of the target depending on trend
4. minProfitDistance: 200 (points) //The minimum distance current ask/bid price must be from the target before a trade can be initiated
5. maxTradesPerLevel: 1 //Suppresses the EA from placing too many trades close together
6. SeparatorDistance: 100 (points) //This is the minimum distance that a new trade must be from an existing trade.
7. HedgingTrendTrade: true //If true the EA will place a hedging trend trade when any other trade reaches its stop loss level
8. HedgingTrendVolume: 2x //multiple of the losing trade's volume
9. HedgingTrendBuffer: 5 (points) //safety buffer when closing hedgingTrendTrade and parent trade
10. hedgingTrendDistance: 400 (points) //the distance to trigger a hedgingTrendTrade
11. numberOfTaps : 2 //The number of additional trades to place if price moves away from the target; these are separated by separatorDistance
12. maxTotalTaps: 10 //The total number of add-on trades permitted at one time
13. MaxTotalTrades: 20 //The total number of trades that can be active at one time
14. ParentVolume: 0.01 (lots) //standard (parent) trade entry volume
15. tapVolume: 1.0 //multiple of the original trade volume)
16. signalBarMultSL: 0.0 //A stop loss that is a multiple of the signal bar height (points)
17. PctSL: 2.0 //% of account balance. Closes any trade when it exceeds 2% of acct.)
18. Failsafe: 1500 //The total drawdown in account currency allowed before the EA exits and removes itself with a notification
19. Breakeven: false //When true if the trade is in profit by a trigger amount place a SL at a certain distance (breakEvenDistance). **Note this doesn't apply to hedgingTrend trades.**
20. BreakEvenTrigger: 40 (points) //Distance price must be in profit before BE is armed
21. BreakEvenDistance: 25 (points) //The distance beyond entry to put the (virtual) SL
22. tradingHours: 01:00 - 11:59 (AM); 12:00 - 23:30 (PM) //can we separate sessions like this or it must be one block of time?

(document continues below)

Flowchart



(document continues below)



[Download the complete flowchart jpg file.](#)

[Download the flowchart PDF file.](#)

(end of document)