



NINJATRADER 8 INDICATOR · NINJAScript, C#

User Guide

SuperTrend signals on closed bars scored by five confluence factors, a stop and three targets, an EMA cloud, a trend tracer, a fourteen-row dashboard with a multi-timeframe row, alerts, and series a strategy can read.

PRODUCT	Hyper Trend PRO for NinjaTrader 8
VERSION	1.0.0
DATE	September 2026
NINJATRADER	8.1.x Desktop, tested on 8.1.8.2
PACKAGE	HyperTrendPro.zip , compiled NinjaScript Add-On
SHA - 256	0923d2e09e0b3391ed26a68a14df289d4fb38d1ae416925463e2d6d26607d651
AUTHOR	Kestutis Balciunas, https://kestutisbalciunas.com/
PRODUCT PAGE	https://kestutisbalciunas.com/ninjatrade/hyper-trend-pro/
SUPPORT	https://kestutisbalciunas.com/contact/

Trading involves substantial risk. Futures, foreign currency and options trading contains substantial risk and is not for every investor. Past, simulated and backtested performance is not necessarily indicative of future results.

Contents

01 What it is

02 Installation

03 Quick start

04 Calculate mode and bar timing

05 Outputs for strategies and the Market
Analyzer

06 Parameters

07 Sessions, time and volume

08 Compatibility

09 Troubleshooting

10 How the port was verified

11 Licensing

12 Support

13 Risk disclosure

What it is

Chart indicator for NinjaTrader 8 Desktop (NinjaScript, C#): a SuperTrend engine whose flips on closed bars become BUY and SELL signals when they reach a confluence score out of five (MACD histogram, EMA 150 against EMA 250, ADX, volume, Hull slope) and pass the optional filters, with a stop and three risk-reward targets fixed at the signal bar, an EMA cloud, a 250-bar trend tracer, momentum-coloured candles, pullback and market-structure labels, a fourteen-row dashboard with a multi-timeframe trend row, and nine alert conditions. Signals only: it never places, modifies or closes orders.

Installation

Requirements

- NinjaTrader 8.1.x Desktop for Windows (tested on 8.1.8.2). Not for NinjaTrader 8.0 or NinjaTrader Web.
- Any data connection that delivers bars for the chart (the indicator needs no Order Flow+ features, no depth, no tick history). Volume is used when the feed supplies it.
- The dashboard's multi-timeframe row loads five minute-based series of the same instrument (5, 15, 30, 60 and 240 minutes) next to the chart's own bars. They load with the chart and need no setup.

Import the Add-On package

1. Control Center > Tools > Import > NinjaScript Add-On.
2. Select `HyperTrendPro.zip` and confirm. NinjaTrader compiles the assembly and reports "Import complete".
3. No restart is needed. If a chart already had an older version, remove and re-add the indicator (a compile does not reload running instances).

Verify installation

1. Open a chart (any instrument, a 1 to 60 minute interval is a good first look), press Ctrl+I.
2. In the Available list find **Hyper Trend PRO**, click add, click OK.
3. The chart shows the SuperTrend line (green under price while bullish, red over it while bearish), candles coloured by momentum, the EMA cloud with its fill, the trend tracer, a BUY or SELL plate on every confirmed flip that passes the filters, the entry, stop and three target lines of the last signal with the risk and reward boxes, the dashboard in the top-right corner and the watermark at the bottom. The longest average is 250 bars, so load at least a day of 1-minute bars before judging the first signals.

Uninstall / update

- Remove: Control Center > Tools > Remove NinjaScript Assembly > Hyper Trend PRO.
- Update: import the new zip over the installed version (NinjaTrader replaces the assembly), then remove and re-add the indicator on every chart that uses it, or reload the workspace.

03

Quick start

First chart

1. Add **Hyper Trend PRO** to a chart (Ctrl+I). Leave every setting at its default.
2. Load enough history: the longest average is the 250-bar trend tracer, so a day of 1-minute bars or a week of 5-minute bars is plenty. The multi-timeframe row of the dashboard needs more, see below.
3. Signals print on closed bars only. Wait for a close that flips the SuperTrend line.

Reading the chart

- **SuperTrend line:** the trend engine. Green under price while the trend is up, red over price while it is down. It is built on the average true range: bands sit **Sensitivity** multiples of a **Tuner**-bar ATR from the mid price and ratchet, and the trend flips when price closes through the far band. Every signal starts as a flip of this line.
- **Candle colours:** by default a momentum gradient from the MACD against the ATR, green for strong upward momentum through yellow, grey and violet to red for strong downward momentum. **13 NinjaTrader > Candle Coloring** switches it to the SuperTrend side, to the confluence score, or off.
- **EMA cloud:** the band between two moving averages, filled green while bullish and red while bearish. **Cloud Style** picks the pair (Smooth, Scalping, Scalping+, Swing).
- **Trend tracer:** a 250-bar EMA, green while the close is above it, red while below.
- **BUY and SELL plates:** a flip that passes the score and the filters gets a three-line plate under the bar for a BUY and above it for a SELL: the banner, the close, and the star rating with the score, for example ★★ (3/5) . Signals never repaint. **Signal Marker Style** can replace the plate with a triangle, an arrow or a dot.
- **Entry, SL, TP1, TP2, TP3:** fixed at the signal bar while **Show TP/SL Area** is on, with a red risk box and a green reward box. The lines and their price tags extend to the right as the chart advances; the levels never move. A target that price has reached gets a tick on its tag. The previous trade's levels stay on the chart until the next signal but one, exactly as in the original.
- **Pullback plates** (▲ PB, ▼ PB, off by default) mark a pullback inside an established trend.
- **Market structure** (off by default) labels confirmed swings HH, HL, LH and LL.

Signal rules in one paragraph

On every bar the indicator scores the bullish and the bearish case out of five: MACD histogram rising or falling, EMA 150 against EMA 250, ADX above 20, volume above its 20-bar average, and the slope of a 55-bar Hull average (the ADX and volume tests count for both sides). A SuperTrend flip on a closed bar becomes a signal when its score reaches the minimum, which is the higher of the **Preset** (All Signals 0, Trend Scalper 2, Quality+ 3, Strong+ 4) and **Min Strength**, and when it passes every optional filter that is switched on: Trending Only, Strong Only, High Vol Only, Cloud-Aligned Only, Contrarian Only. The stop is $\text{ATR(Tuner)} \times \text{Sensitivity}$ from the entry (or a percentage with **Use % SL**), and the three targets are **TP1/TP2/TP3 (RR)** multiples of that same risk.

The dashboard

Fourteen rows in the corner chosen by **10 Dashboard > Dash Position**. It summarises the current bar plus the last signal; the trade rows freeze at the last signal until the next one.

ROW	LEFT	RIGHT
1	the product name	the product name and version
2	MTF TREND	section header
3	Timeframes	an arrow per timeframe: M5, M15, M30, H1, H4
4	ANALYSIS	section header
5	Market State	Strong Trend above ADX 40, Trending above 20, otherwise Ranging, with the ADX value
6	Volatility	High above 1.5 percent of price, Normal above 0.5, otherwise Low, with the figure
7	Volume	Spike above twice the 20-bar average, Above Avg over it, otherwise Below Avg, with the ratio
8	Session	Tokyo, London, New York or Sydney, by UTC hour
9	Sentiment	a second score out of five: trend direction, the EMA pair, the Hull slope, the MACD histogram and the multi-timeframe majority
10	SIGNAL	section header
11	Direction	LONG, SHORT or WAITING
12	Entry / SL	the two prices of the tracked trade
13	TP1/TP2/TP3	the three targets, each with a tick once price has reached it
14	the product name	the product name and version

The multi-timeframe row. Each arrow answers one question for its timeframe: is the close above its 200-bar EMA? It is read from the last CLOSED bar of that timeframe, so it does not flicker within a candle. A timeframe with fewer than 201 closed bars loaded reads down: on a chart with a few days of data the H4 arrow, and sometimes H1, stays down until more history is loaded (Data Series > Days to load; about 40 days cover H4). The row is green when four or five agree upward, red when four or five agree downward, and grey in between. It feeds the Sentiment score and nothing else: no signal, level or plot depends on it.

A tick beside a target records that price traded through that level on the loaded history. There is no spread, commission or slippage in it, and it is not a filled order.

Alerts

Nine events, each behind its own toggle in **12 Alerts**, evaluated for bars that close after the indicator went live (never during the history pass, and never for a bar that closed more than two bar periods ago). They land in the Alerts Log with an optional sound (`Sound File` inside NinjaTrader's sounds folder) and an optional email through NinjaTrader's Mail share service.

TOGGLE	FIRES WHEN
Alert: Buy Signal	a confirmed BUY signal
Alert: Sell Signal	a confirmed SELL signal
Alert: Any Signal	either of the two above
Alert: Strong Signal	a signal whose score is 4 or 5
Alert: SL Hit	price reaches the stop of the tracked trade
Alert: TP1 Hit	price reaches the first target for the first time
Alert: TP2 Hit	the second target
Alert: TP3 Hit	the third target, which also closes the tracked trade
Alert: Trend Change	the SuperTrend flips, whether or not the flip passed the filters

All nine are on by default, so one strong BUY raises three lines (Buy, Any, Strong); switch off the ones you do not want. The buy and sell bodies are the original's text with the entry, stop and targets, or the same values as JSON with **Webhook JSON Format** on. Nothing is sent anywhere by the indicator itself.

Using the values in a strategy or the Market Analyzer

Eight series are exposed: `CloudUpper`, `CloudLower`, `SuperTrend`, `TrendTracer`, `TrailingSL`, and the hidden `Signal` (+1 on a BUY bar, -1 on a SELL bar, 0 otherwise), `TradeDirection` (+1 while a long trade is tracked, -1 for a short, 0 when flat) and `SignalScore` (the 0 to 5 score on a signal bar). Example: `if (HyperTrendPro(...).Signal[0] > 0)`. A strategy that hosts the indicator loads the same five minute series.

Calculate mode and bar timing

- Default `Calculate.OnBarClose`. Signals are evaluated on closed bars only (Pine `barstate.isconfirmed`).
- `OnEachTick` / `OnPriceChange`: the forming bar is previewed on every tick (engine `Step` with `confirmed = false`) and that step is undone before the next one, exactly as TradingView rolls the script back before each realtime tick; level hits and the dashboard follow the forming bar, signals wait for the close. The bar is committed once at `IsFirstTickOfBar` of the next bar. Signals never repaint.
- Historical pass: one `OnBarUpdate` per closed bar. A still-forming last historical bar is previewed, not committed. The engine slot is derived from `CurrentBar`, so the historical to realtime transition can neither double-feed nor skip a bar; a skipped range is refilled from the series.
- The draw model is put on the chart when the newest bar is reached and on every bar after that, not during the history pass, which keeps a 20-day 1-minute load fast.
- Tick Replay: handled as per-tick realtime.
- The indicator keeps calculating while its chart window is minimized (`IsSuspendedWhileInactive = false`), and an alert is only raised for a bar that closed within the last two bar periods.

Outputs for strategies and the Market Analyzer

PLOT / SERIES	VALUES
<code>CloudUpper</code> , <code>CloudLower</code> (plots 0, 1)	the two edges of the EMA cloud; NaN while hidden
<code>SuperTrend</code> (plot 2)	the SuperTrend line, coloured by its side per bar
<code>TrendTracer</code> (plot 3)	the 250-bar EMA, coloured by the close's side per bar; NaN while hidden
<code>TrailingSL</code> (plot 4)	the one-way ATR trail of an open trade; NaN while hidden or flat
<code>Signal</code> (plot 5, hidden)	+1 on a confirmed BUY bar, -1 on a confirmed SELL bar, 0 otherwise
<code>TradeDirection</code> (plot 6, hidden)	+1 while a long trade is tracked, -1 for a short, 0 when flat
<code>SignalScore</code> (plot 7, hidden)	the 0 to 5 confluence score of the signal on a signal bar, 0 otherwise

Example in a strategy: `var htp = HyperTrendPro(...); if (htp.Signal[0] > 0) ...` (the accessor is generated by NinjaTrader on compile).

Parameters

The 37 inputs of the original script keep their titles, groups, order and defaults (the group names carry a number prefix because NinjaTrader sorts groups alphabetically). The 26 NinjaTrader-only additions (the original exposes its nine alert conditions through TradingView's alert dialog) are the nine per-event alert toggles and the sound and email settings in group 12 Alerts, and the candle colouring, label glyphs, signal marker style, label size, transparency and offset, the shared label cap, the three Use TP switches, the Strong Only rule and the two diagnostic switches in group 13 NinjaTrader: 63 parameters in total.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
01 Appearance	Bull Color	colour	HtpNtColors.Frozen(HtpPalette.BullDefault)		BullBrush	
01 Appearance	Bear Color	colour	HtpNtColors.Frozen(HtpPalette.BearDefault)		BearBrush	
01 Appearance	Show Signals	bool	true		ShowSignals	
02 SuperTrend Engine	Sensitivity (ATR Factor)	double	3.1	0.1,double.MaxValue	Sensitivity	
02 SuperTrend Engine	Tuner (ATR Length)	int	25	1,int.MaxValue	Tuner	
03 Signal Filtering	Preset	HtpPreset	AllSignals		Preset	
03 Signal Filtering	Min Strength (0-5)	int	0	0 to 5	MinStrength	
03 Signal Filtering	Trending Only (ADX>20)	bool	false		TrendOnly	
03 Signal Filtering	Strong Only	bool	false		StrongOnly	
03 Signal Filtering	High Vol Only	bool	false		HighVolOnly	
03 Signal Filtering	Cloud-Aligned Only	bool	false		CloudAligned	
03 Signal Filtering	Contrarian Only	bool	false		ContrarianOnly	
04 Pullback	Show Pullback Signals	bool	false		ShowPullback	
04 Pullback	Pullback Tuner	int	5	1,int.MaxValue	PbTuner	
05 Contrarian RSI	Contrarian Length	int	50	5,int.MaxValue	ContLen	
05 Contrarian RSI	Smoothing	int	30	1,int.MaxValue	ContSmooth	
05 Contrarian RSI	Overbought	int	60	50 to 95	ContOB	

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
05 Contrarian RSI	Oversold	int	40	5 to 50	ContOS	
06 EMA Cloud	Show Cloud	bool	true		ShowCloud	
06 EMA Cloud	Cloud Style	HtpCloudStyle	Smooth		CloudStyle	
07 Trend Tracer	Show Tracer	bool	true		ShowTracer	
07 Trend Tracer	Tracer Length	int	250	10,int.MaxValue	TracerLen	
08 TP / SL	Show TP/SL Area	bool	true		ShowTPSL	
08 TP / SL	Show Trailing Stop	bool	false		ShowTrailing	
08 TP / SL	Use % SL	bool	false		UsePctSL	
08 TP / SL	% SL	double	1.0	0.1,double.MaxValue	PctSL	
08 TP / SL	TP1 (RR)	double	1.0	0.1,double.MaxValue	Tp1	
08 TP / SL	TP2 (RR)	double	2.0	0.1,double.MaxValue	Tp2	
08 TP / SL	TP3 (RR)	double	3.0	0.1,double.MaxValue	Tp3	
09 Market Structure	Show Swings	bool	false		ShowSwings	
09 Market Structure	Swing Period	int	10	3,int.MaxValue	SwingPeriod	
10 Dashboard	Show Dashboard	bool	true		ShowDash	
10 Dashboard	Dash Position	HtpDashPos	TopRight		DashPos	
10 Dashboard	Dash Size	HtpDashSize	Small		DashSize	
11 Watermark	Show Watermark	bool	true		ShowWatermark	
12 Alerts	Enable Alerts	bool	true		EnableAlerts	
12 Alerts	Webhook JSON Format	bool	false		WebhookJson	
12 Alerts	Alert: Buy Signal	bool	true		AlertBuy	
12 Alerts	Alert: Sell Signal	bool	true		AlertSell	
12 Alerts	Alert: Any Signal	bool	true		AlertAny	
12 Alerts	Alert: Strong Signal	bool	true		AlertStrong	
12 Alerts	Alert: SL Hit	bool	true		AlertSl	
12 Alerts	Alert: TP1 Hit	bool	true		AlertTp1	

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
12 Alerts	Alert: TP2 Hit	bool	true		AlertTp2	
12 Alerts	Alert: TP3 Hit	bool	true		AlertTp3	
12 Alerts	Alert: Trend Change	bool	true		AlertTrend	
12 Alerts	Play Sound	bool	false		PlaySound	
12 Alerts	Sound File (in NinjaTrader's sounds folder)	string	"Alert1.wav"		AlertSoundFile	
12 Alerts	Send Email	bool	false		SendEmail	
12 Alerts	Email To	string	""		EmailTo	
13 NinjaTrader	Candle Coloring	HtpCandleMode	Momentum		CandleMode	
13 NinjaTrader	Label Glyphs	HtpGlyphMode	Symbols		GlyphMode	
13 NinjaTrader	Signal Marker Style	HtpMarkerStyle	Label		MarkerStyle	
13 NinjaTrader	Signal Label Size	HtpLabelSizeMode	Auto		LabelSize	
13 NinjaTrader	Signal Label Transparency (0-100)	int	0	0 to 100	LabelOpacity	
13 NinjaTrader	Signal Label Offset (ATR)	double	0.0	0.0,double.MaxValue	LabelOffsetAtr	
13 NinjaTrader	Label Cap (all labels share one pool)	int	500	1 to 5000	LabelCap	
13 NinjaTrader	Use TP1	bool	true		UseTp1	
13 NinjaTrader	Use TP2	bool	true		UseTp2	
13 NinjaTrader	Use TP3	bool	true		UseTp3	
13 NinjaTrader	Strong Only Rule	HtpStrongRule	Score		StrongRule	
13 NinjaTrader	Debug Mode	bool	false		DebugMode	
13 NinjaTrader	Dump Bars + Outputs CSV	bool	false		DebugDumpCsv	

63 parameters.

Sessions, time and volume

Sessions

The indicator has no session logic of its own: every bar the chart delivers is fed to the engine in order, so the chart's Trading Hours template decides which bars exist. With an ETH template (for example `CME US Index Futures ETH`) the overnight bars take part in the ATR, the averages and the volume test; with an RTH template they do not. Both are valid; the values differ, as they do for any indicator.

The dashboard's **Session** row is a label only: it names Tokyo, London, New York or Sydney from the UTC hour of the current bar, the original's rule, and filters nothing.

Time stamps

NinjaTrader stamps a bar with its CLOSE time in the platform time zone. The engine's bar open time (used for the Session row, the alert de-duplication and the CSV dump) is reconstructed as close time minus the bar period and converted to UTC; for non-time bars the observed bar spacing is used. Level lines and boxes are anchored at bar times; the part that lies past the current bar is positioned on the bar period, so on tick, volume or range bars its x position is approximate.

The five extra series

The multi-timeframe row reads 5, 15, 30, 60 and 240-minute bars of the chart's instrument, built by NinjaTrader from the same data and the same Trading Hours template as the chart. Only CLOSED bars of those series enter their 200-bar EMA, so the row never changes within a candle of its timeframe. A timeframe reads down until 201 of its bars have closed inside the loaded window: with 5 days loaded M5 to M30 are live, H1 needs about 9 trading days and H4 about 40.

Volume

`Volume[]` on a futures chart is real contract volume. The original script used `volume`, which on forex feeds is tick volume; on NinjaTrader futures the volume factor of the score and the High Vol Only filter therefore compare exchange volume against its own 20-bar average, a better input for the same test.

Contract rollover

A continuous contract (merge policy `MergeBackAdjusted` or `MergeNonBackAdjusted` in the Data Series dialog) shows a price jump at the roll unless back-adjusted. The ATR, the SuperTrend bands, the stop and the targets are price differences, so a non-back-adjusted roll produces one oversized true range and can flip the trend on the roll bar; with back adjustment there is none. State the merge policy when comparing values across machines.

Compatibility

ITEM	STATUS
NinjaTrader version	8.1.x Desktop (tested 8.1.8.2). Not 8.0, not NinjaTrader Web or mobile.
Product type	Indicator, price panel overlay. Multi-series: the chart's own bars plus five minute series (5, 15, 30, 60, 240) of the same instrument, used only by the dashboard's multi-timeframe row.
Instruments	Any instrument with bars: futures, micro futures, forex, stocks, crypto CFDs. Verified on MNQ (CME) 1-minute; the engine was verified on EURUSD M5 (stage 1 parity).
Bar types	Time-based (second, minute, day, week, month) fully; tick, volume, range, Renko and other non-time bars run, with the alert period string and the right-hand part of the level lines derived from the observed bar spacing.
Calculate modes	OnBarClose (default), OnEachTick, OnPriceChange (forming-bar preview that is undone before the next step; signals on closed bars only). Tick Replay handled as per-tick.
Multi-instance	Yes: every draw object and alert id carries the instrument, period and instance hash.
Market Analyzer	Loads (no chart drawing); <code>Signal</code> , <code>TradeDirection</code> , <code>SignalScore</code> and the five visible series are available as columns. Not advertised as a scanner.
Strategy Analyzer / Strategy Builder	The eight series can be referenced from a strategy; all inputs are <code>[NinjaScriptProperty]</code> . The host strategy loads the same five minute series.
Data requirements	Bars only. Volume is used by the volume factor when present. The multi-timeframe row needs 201 closed bars per timeframe (about 40 days for H4) and reads down until then.
Rendering	Plots and draw objects through the standard NinjaTrader API; the dashboard, the watermark and the cloud fill through SharpDX in <code>OnRender</code> , with DPI scaling and full resource cleanup on render-target change and termination.
Draw objects	One shared pool of at most Label Cap labels (500 by default, oldest removed first), five level lines, five price tags and two boxes per tracked trade, two trades kept.
Dark and light charts	The dashboard uses the original's fixed dark palette on either background.
Templates and workspaces	All settings serialize (colours through the standard brush serializer).
Minimized chart	Keeps calculating and alerting while its chart window is minimized (<code>IsSuspendedWhileInactive</code> off). A bar that closed more than two bar periods ago is never alerted, so a reconnect or a catch-up never replays old alerts.
Order handling	None. The indicator never touches orders, positions or accounts.

Troubleshooting

SYMPTOM	CAUSE AND FIX
No signals	A signal needs a SuperTrend flip on a closed bar AND a score that reaches the higher of Preset and Min Strength AND every optional filter that is on. Set the preset to All Signals and the filters off to see every flip, then tighten.
The H4 (or H1) arrow of the dashboard always points down	That timeframe has fewer than 201 closed bars loaded. Raise Data Series > Days to load (about 40 days cover H4). Nothing but the Sentiment score depends on that row.
The chart loads slower than with other indicators	Five extra minute series load with it for the multi-timeframe row. The load time follows Days to load.
Signals print but no levels	08 TP / SL > Show TP/SL Area is off. In that state the original script tracks no trade, and this build does the same.
Two sets of level lines on the chart	By design: the original deletes a trade's drawing two signals later, so the previous trade stays visible next to the current one.
Older labels disappear	All labels share one pool of Label Cap objects (500 by default); the oldest goes first, whatever its kind. Raise the cap in group 13 if you want more history labelled.
A label covers a dashboard cell	NinjaTrader paints draw objects above custom rendering. Move the dashboard with Dash Position or add right margin (chart Properties > Right side margin).
Dashboard missing	10 Dashboard > Show Dashboard on? The dashboard is drawn in the price panel; with the indicator in its own panel it is not visible.
Dashboard text tiny or huge	Sizes follow the Windows display scaling and Dash Size .
Level price tags cut off at the right edge	They sit to the right of the current bar while the trade is open; add right margin (chart Properties > Right side margin).
Candles keep the platform's colours	13 NinjaTrader > Candle Coloring is Off, or the Trend / Confluence mode is still in warm-up.
Alerts do not fire	Alerts fire only for bars that close after the chart loaded, only in realtime, and only with Enable Alerts plus the per-event toggle on. Email needs the Mail share service configured under Tools > Options > Share.
Three alerts for one signal	Buy (or Sell), Any and Strong are separate events, all on by default. Switch off the ones you do not want.
Sound not played	Sound File must name a file inside <code><NinjaTrader 8 install>\sounds\</code> .
Values differ from another platform on the same symbol	Bars differ by feed and session template; compare on identical bars (see the chapter on how the port was verified).
An error line names Hyper Trend PRO in the Log tab	Set 13 NinjaTrader > Debug Mode on, reproduce, and send <code>Documents\NinjaTrader 8\ntp_nt_debug.log</code> with the chart interval, instrument and NinjaTrader version.
Two dashboards on one chart	Two instances are configured (Ctrl+I, Configured list). Remove one.

How the port was verified

The verification has two stages, which measure different things and are reported separately. It is never stated as "verified against TradingView on futures bars": TradingView's CME feed and NinjaTrader's feed do not carry the same bars (sessions, rollover, continuous-contract construction), so there is no such comparison to make.

Stage 1: engine vs the TradingView original (cTrader edition, 12 September 2026)

The platform-free engine that ships here is the same code as the cTrader edition's engine apart from its namespace line. That engine was replayed on 20,141 identical ICMARKETS:EURUSD M5 bars read from the TradingView chart model and compared with probe copies of the published Pine script:

- 15,991,469 comparisons across 34 runs, 0 mismatches: four probe generations bar by bar (1,848,171 comparisons), 26 input sweeps covering the presets, every filter, the cloud styles, the pullback tuner, the percent stop, the target ratios, the tracer, the trailing stop and the swing period (14,128,632), and four drawn-object runs comparing every label, line, box and dashboard cell by its properties (14,666);
- declared and excluded from that zero: 20 intended differences, all brand strings (dashboard header and footer, their version strings and the watermark), and 19 tail-residue labels at the trim boundary of the shared 500-label pool, each proven older than every label the two sides agree on.

Stage 2: NinjaTrader host vs the engine on identical NinjaTrader bars (19 September 2026)

With **Dump Bars + Outputs CSV** on, the indicator writes every committed bar (time, open, high, low, close, volume) together with the engine outputs as the host saw them, in exactly the columns the cTrader parity harness writes: 105 series covering both SuperTrend state machines, the MACD, the EMA pair, the DMI and ADX, the volume average, the Hull averages, both confluence scores, the contrarian RSI chain, the WaveTrend chain and the pullback flags, all four cloud styles, the tracer, the preset and filter gates, the signals, the whole trade state machine (entry, stop, three targets, hit flags, trailing stop, an integer digest of the trade state), the candle colour and the market structure. The same bars are replayed through a build of the identical engine outside NinjaTrader, with no host in between, and the two files are diffed value by value.

Result on MNQ DEC26, 1-minute bars, NinjaTrader 8.1.8.2, Simulation connection (delayed CME data), Calculate on bar close, 13 to 18 September 2026:

bars compared	6,899
bar identity (time)	0 mismatching rows
series compared	105
cells compared	724,395
mismatches	0

max abs difference	0
SuperTrend flips on the run	77 up, 78 down
signals on the run	77 buy, 78 sell, on the same bars
first-target hits / stop hits on the run	66 / 14, on the same bars

What stage 2 proves: the host feeds the engine exactly one closed bar per chart bar with the right OHLCV and the right UTC open time, never double-feeds or skips a bar, and plots the values the engine produced. The five extra minute series feed the dashboard's multi-timeframe row only and enter none of the 105 series. What it does not prove: anything about future performance.

Stage 2b: the same check across the historical-to-realtime transition (22 September 2026)

The run above was replayed on a closed market. This one was taken on a trading day with **Calculate** set to **OnEachTick**: the chart loaded its history, then stayed live for 21 minutes of MNQ DEC26 1-minute bars (19:27 to 19:48 platform time), with the forming bar previewed on every tick and committed once at the first tick of the next bar. Every committed bar went into the dump, the file was closed, replayed through the engine outside NinjaTrader and diffed value by value.

bars compared	5,258 (history plus the live bars)
bar identity (time)	0 mismatching rows
series compared	105
cells compared	552,090
mismatches	0
max abs difference	0

trace in the committed bars, the historical-to-realtime boundary neither double-feeds nor skips a bar, and the values plotted for live bars equal the engine's.

11

Licensing

License model

The product is distributed as a compiled NinjaScript Add-On and licensed through NinjaTrader's user-based Vendor Licensing (NinjaTrader 8.1.5.0 or later). A licence is tied to the customer's NinjaTrader account login, not to a machine: the same login works on every installation of NinjaTrader Desktop the customer uses, and no Machine ID has to be sent to the vendor. Licences are created in the NinjaTrader Ecosystem Vendor Dashboard against the customer's NinjaTrader account email address, with a licence type and an expiry date.

The indicator carries the `VendorLicense(productId)` call in its constructor, with the Product ID of its add-on in the Vendor Dashboard. NinjaTrader verifies the licence whenever the indicator is used.

No free trial

The product is sold as a paid licence only; no free trial is offered. A licence is issued to the buyer's NinjaTrader account email after payment, normally within a few hours.

Activation and transfer

Nothing to activate: log in to NinjaTrader with the licensed account. A new PC needs no transfer. NinjaTrader allows two simultaneous market-data sessions per account, which is what prevents a licence from being shared.

Behavior without a valid licence

NinjaTrader shows its own licence notice (invalid or expired) and stops calling the indicator's calculation and plot methods: the indicator draws nothing and raises no alerts, while its settings still load. It never places orders (it has no order code) and exposes no proprietary data. A licensing-server outage is handled by NinjaTrader's verification (it retries against a secondary server) and must never crash the chart.

Licensing is a practical access-control layer, not a guarantee against copying.

12

Support

Questions go through the contact page at <https://kestutisbalciunas.com/contact/> with the NinjaTrader version, the instrument and the chart interval. For an error, switch **11 NinjaTrader > Debug Mode** on, reproduce it once and attach `http_nt_debug.log` from the `Documents/NinjaTrader 8` folder; switch Debug Mode off afterwards.

The product page, with the documentation and the change log, is <https://kestutisbalciunas.com/ninjatrade/hyper-trend-pro/>. Updates and refunds follow the terms of the NinjaTrader listing once it is live.

13

Risk disclosure

Trading involves substantial risk. Past, simulated, and backtested performance is not necessarily indicative of future results.

Futures, foreign currency and options trading contains substantial risk and is not for every investor. An investor could potentially lose all or more than the initial investment. Risk capital is money that can be lost without jeopardizing one's financial security or lifestyle. Only risk capital should be used for trading and only those with sufficient risk capital should consider trading.

Hyper Trend PRO is a technical analysis tool. It draws trend lines, levels and signals computed from historical bars; it does not place, modify or close orders and it gives no trade advice. The star rating describes how many of the enabled filters agreed on the breakout bar; it is a description of agreement, not a probability, and not a forecast. Nothing in the software or its documentation is a promise of returns, win rates or funded-account success.

NinjaTrader is a registered trademark of NinjaTrader Group, LLC. No NinjaTrader company has any affiliation with the owner, developer, or provider of the products or services described herein, or any interest, ownership or otherwise, in any such product or service, or endorses, recommends or approves any such product or service.