

COPY-PASTE KIT — CLAUDE + TRADINGVIEW VIDEO

How to use this doc: Keep it open on a second monitor during filming. Every code block below is ready to copy straight into Claude Code, Terminal, or your settings file. Nothing to modify, nothing to guess at.

1. THE MASTER SETUP PROMPT

When to use: The setup section of the video (Section 4). This is the single prompt viewers paste to install everything.

Paste into: Claude Code

None

You are going to install the TradingView MCP server and configure it for a crypto trading workflow. Follow these steps in order. At each step, if something fails, report the error clearly and stop – do not continue past a failure.

Step 1 – Install the MCP server. Clone

<https://github.com/tradesdontlie/tradingview-mcp.git> to ~/tradingview-mcp. Run npm install inside it. If the directory already exists, pull the latest changes instead of cloning.

Step 2 – Add the MCP to my Claude Code config. Edit ~/.claude/.mcp.json to add the tradingview server. If the file already exists and has other MCP servers, merge the tradingview entry into the existing mcpServers object – do not overwrite other servers. The entry to add:

```
{
  "mcpServers": {
    "tradingview": {
      "command": "node",
      "args": ["<HOME>/tradingview-mcp/src/server.js"]
    }
  }
}
```

Replace <HOME> with the actual absolute path to my home directory.

Step 3 – Create my trading rules file at ~/tradingview-mcp/rules.json with this crypto swing-trading configuration:

```
{
  "watchlist": {
    "majors": ["BINANCE:BTCUSDT", "BINANCE:ETHUSDT", "BINANCE:SOLUSDT"],
    "alts": ["BINANCE:LINKUSDT", "BINANCE:AVAXUSDT", "BINANCE:SUIUSDT"],
    "macro": ["CRYPTOCAP:TOTAL", "CRYPTOCAP:TOTAL3", "CRYPTOCAP:BTC.D"]
  },
  "timeframes_to_check": ["1W", "1D", "4H"],
  "bias_criteria": {
    "bullish": "Price above 50D EMA, RSI on daily between 45 and 70, higher highs and higher lows on 4H",
    "bearish": "Price below 50D EMA, RSI on daily below 45, lower highs and lower lows on 4H",
    "neutral": "Price chopping around 50D EMA, RSI between 40 and 60, no clear structure"
  },
  "risk_rules": {
    "max_risk_per_trade": "1% of portfolio",
    "min_rr_ratio": 2,
    "no_trades_during": ["major US CPI", "FOMC", "weekend thin liquidity"]
  },
  "indicators_i_care_about": ["RSI (14)", "MACD (12, 26, 9)", "50 EMA", "200 EMA", "Volume"]
}
```

Step 4 – Launch TradingView with the debug port using the tv_launch tool. If tv_launch isn't available yet, auto-detect the TradingView Desktop app on this machine and launch it with the --remote-debugging-port=9222 flag. Mac path: /Applications/TradingView.app/Contents/MacOS/TradingView. Windows path: %LOCALAPPDATA%\TradingView\TradingView.exe. Linux path: /opt/TradingView/tradingview.

Step 5 – Verify the connection by running tv_health_check. If it returns cdp_connected: true, the setup is working.

Step 6 – Once health check passes, fetch the current price of BTCUSDT and report it to me. Then report back the full setup status:

- MCP installed and connected: yes/no
- Rules file created at: [path]
- TradingView connected on port 9222: yes/no
- Current BTC price: [number]
- Ready to use: yes/no

If all five items pass, the setup is complete.

2. PRE-APPROVE TRADINGVIEW TOOLS (STOP PERMISSION PROMPTS)

When to use: Right after setup. Tell viewers this is optional but highly recommended.

Paste into: Claude Code

None

Edit my Claude Code settings to pre-approve all TradingView MCP tools so I stop getting permission prompts for them.

Open ~/.claude/settings.json. If the file doesn't exist, create it. If it exists and already has a permissions.allow array, merge the new entries into it – do not overwrite any existing rules.

Add this entry to the allow list:

mcp__tradingview__*

Final file should look like this (merging with anything already there):

```
{
  "permissions": {
    "allow": [
      "mcp__tradingview__*"
    ]
  }
}
```

After saving, confirm the file contents back to me. Then tell me to restart Claude Code for the new permissions to take effect.

Reminder: viewers must fully quit and restart Claude Code after this for it to take effect.

3. VERIFICATION COMMANDS

When to use: Setup confirmation on camera.

Paste into: Claude Code

None

```
tv_health_check
```

Expected output: { "success": true, "cdp_connected": true, "chart_symbol": "...", "api_available": true }

If `cdp_connected: false`, TradingView isn't running with the debug port. Quit TradingView fully and relaunch via `tv_launch`.

4. DEMO PROMPTS

DEMO 1 — THE ANALYST (Section 5)

Setup before filming: Open a blank BTC chart, 4-hour timeframe, no indicators.

Prompt:

None

```
Analyse this BTC chart and give me the best long and short opportunities right now. If you find good setups, ask me whether I want you to draw them on the chart.
```

Backup prompts (if the first read comes back weak):

None

```
Look at my current chart. Identify the key support and resistance levels, the market structure, and any clear trade setups. Then offer to draw your analysis directly onto the chart.
```

None

```
I want a full technical breakdown of whatever chart I have open right now. Market structure, key levels, potential entries and invalidations. Then draw your findings on the chart.
```

DEMO 2 — THE BUILDER (Section 6)

Setup before filming: BTC chart open, any timeframe with visible history.

Prompt:

None

```
Build me an indicator that prints a buy label when RSI crosses up through 30, and a sell label when RSI crosses down through 70, but only when price is on the right side of the 200 EMA. Write the Pine Script, inject it into TradingView, compile it, fix any errors, and save it to my account.
```

Follow-up prompt (for the "tweak it" moment on camera):

None

```
Now modify that indicator so it only fires signals when volume is above the 20-period average. Recompile and save.
```

DEMO 3 — THE ASSISTANT (Section 7)

Setup before filming: BTC 4-hour chart with your normal indicator stack. Pick a moment where there's a visible potential setup.

Prompt:

None

```
Given the current chart, where's the cleanest invalidation if I long here? And what's the first level of resistance I'd take partial profit on? Use the actual indicators on my chart, not general market commentary.
```

Backup prompts (for variety if the first doesn't land well):

None

```
Looking at the current state of this chart, what's the highest-probability trade setup right now? Give me entry, stop, and target with reasoning grounded in what's visible.
```

None

If I'm bearish this chart, walk me through the cleanest short setup given what's on screen right now. Include where I'd be wrong.

DEMO 4 — THE AUTOMATOR (Section 8)

Setup before filming: Make sure your crypto watchlist exists in TradingView. Create a local folder for the screenshots in advance (e.g. `~/Desktop/setups-tonight`).

Prompt:

None

Scan my top ten crypto majors on the daily timeframe. Find the three with the cleanest bullish market structure – higher highs, higher lows, price above the 50 EMA. For each of those three, set a price alert at the most recent swing high. Take a screenshot of each chart and save it to `~/Desktop/setups-tonight`. At the end, give me a summary list of the three symbols, their levels, and confirm the alerts are live.

If this takes too long on camera: trim to "scan my top five majors" instead of ten.

5. PHONE FINALE — TELEGRAM VOICE PROMPTS (Section 10)

When to use: The phone screen-record shot outside.

Voice prompts to try (record multiple, pick the best take):

None

Oscar – anything interesting in the majors right now?

None

Oscar, give me a quick read on BTC and ETH right now.

None

Oscar, what setups are triggering on my watchlist today?

Rules for the voice take:

- Sound natural, not scripted — like you're asking a colleague
 - Keep each prompt under 10 seconds of audio
 - Film 3-4 takes with slightly different questions
 - Pre-warm Oscar so the response is fast
-

6. SETTINGS.JSON SNIPPET (standalone for the Telegram kit)

Drop this file into Crypto Edge Telegram alongside the master prompt.

File path: `~/ .claude/settings.json`

```
JSON
{
  "permissions": {
    "allow": [
      "mcp__tradingview__*"
    ]
  }
}
```

7. TROUBLESHOOTING LINES (for the honest-review section)

Keep these handy in case something goes wrong during a live demo and you need to cover:

- **Health check fails:** "TradingView didn't launch with the debug port — close it fully, use the tv_launch tool, try again."
- **MCP returns nothing:** "Sometimes the MCP needs a Claude Code restart after install. Restart and re-run."
- **Demo is slow:** "This one's going to take a minute — I'll fast-forward the boring part in editing." (Normalises the wait for viewers.)
- **Prompt returns a weak answer:** "That one didn't give me much — let me try a different angle." (Re-prompt with a backup.)

8. THE DESCRIPTION LINKS (for video description)

Fill these in before publishing:

- Crypto Edge Telegram: [LINK]
 - Original MCP by @tradesdontlie: <https://github.com/tradesdontlie/tradingview-mcp>
 - TradingView Desktop download: <https://www.tradingview.com/desktop/>
 - Claude Code install: <https://www.claude.com/claude-code>
 - Next video — \$20K AI trading bot: [LINK when live]
-

QUICK REFERENCE CARD

Paste this taped to your monitor on filming day:

Moment in script	What to paste
Section 4 (Setup)	Master setup prompt (Section 1 above)
Section 4 (tip)	Pre-approval prompt (Section 2 above)
Section 5 (Analyst)	Demo 1 prompt
Section 6 (Builder)	Demo 2 prompt + follow-up
Section 7 (Assistant)	Demo 3 prompt
Section 8 (Automator)	Demo 4 prompt
Section 10 (Phone)	Voice prompt 1, 2, or 3

Done.