A Technical Blueprint for Implementing Dr. Eric Wish's Trading Methodologies in Python

System Architecture and Project Blueprint

This document outlines a comprehensive architectural and implementation plan for a Python-based stock screening application designed to codify the trading strategies of Dr. Eric Wish. The system is engineered to translate his market philosophy and specific technical setups into a robust, automated, and modular program. The primary objective is to screen a universe of equities for high-probability trading signals, contingent upon a favorable overall market environment as determined by a proprietary market timing model.

Core Philosophy: A Hierarchical Screening Model

The foundational principle of Dr. Wish's methodology is the primacy of the general market trend.[1] Individual stock setups, regardless of their technical perfection, are considered predisposed to failure within a broader market decline or bear market.[2] This philosophy dictates that a trader's primary task is not to predict the market but to react to its prevailing direction, playing offense in uptrends and defense in downtrends.[3] The system's architecture is a direct codification of this principle, employing a hierarchical, top-down screening process.

This structure is more than a matter of computational efficiency; it is the implementation of a core risk management discipline. By programmatically requiring a "green light" from the market timing model before any individual stock analysis can occur, the system enforces a rule that discretionary traders often find difficult to maintain: the discipline to remain on the sidelines during periods of elevated market risk. The architecture transforms a strategic guideline into an immutable system constraint, automating not just the identification of signals but also the higher-order rule of selective engagement.

The system's operational logic will be executed in two distinct stages:

1. **Stage 1: Market Regime Filtering.** The initial and most critical step involves the execution of the General Market Index (GMI) module. This module will analyze the

health of a major market index (e.g., the Nasdaq 100 via the QQQ ETF) and produce a binary determination of the market's health. This output, conceptualized as "GMI Green" (favorable) or "GMI Red" (unfavorable), will function as a master switch for the entire screening process.

2. **Stage 2: Individual Security Screening.** The system will only proceed to this stage if the GMI module returns a favorable "GMI Green" signal. Upon receiving this confirmation, the application will iterate through a user-defined list of individual stock tickers. For each ticker, it will apply the algorithmic logic for the Green Line Breakout (GLB), Blue Dot, and Black Dot signals. Stocks that meet the criteria for any of these setups will be collected for the final report. If the GMI signal is "Red," this stage is bypassed entirely, and the system concludes its run, effectively preventing the generation of trade ideas in a hostile market environment.

Proposed Technical Stack and Data Structures

To ensure robustness, maintainability, and efficiency, the project will be built upon a standard, well-supported stack of Python libraries widely used in quantitative finance and data analysis.

- **Programming Language:** Python 3.x will be used for its extensive ecosystem of data science and financial analysis libraries.
- **Core Libraries:**
  - **pandas:** This library will be central to the application's data handling. All historical price data (Open, High, Low, Close, Volume - OHLCV) will be loaded into and manipulated via pandas DataFrames. Its powerful time-series capabilities, such as resampling and rolling window calculations, are essential for implementing the required indicators.
  - **NumPy:** As a dependency of pandas, NumPy will provide the underlying engine for high-performance numerical computations and array manipulations, ensuring that calculations are performed efficiently.
  - **ta / talib-wrapper:** To accelerate development and guarantee the accuracy of standard technical indicator calculations, a dedicated technical analysis library such as ta or a wrapper for the industry-standard TA-Lib (talib-wrapper) will be employed. These libraries provide pre-built, optimized functions for indicators like Simple Moving Averages (SMA), Exponential Moving Averages (EMA), and the Stochastic Oscillator, which are fundamental components of Dr. Wish's models.
- **Data Input:** The system will be designed to operate on locally stored historical data, as specified in the project requirements. It will expect daily OHLCV data for a universe of stock tickers, likely in a common format such as Comma-Separated Values (CSV). Each file should be named according to its ticker symbol (e.g., AAPL.csv) for straightforward processing.

Project Workflow Diagram

The logical flow of data and decision-making within the application can be visualized as follows. This workflow ensures that the hierarchical screening philosophy is strictly adhered to at every stage of execution.

1. **Start Execution:** The main script is initiated.
2. **Load Market Index Data:** The system loads the historical OHLCV data for the designated market proxy, such as the QQQ ETF.
3. **Calculate GMI Status:** The GMI module is called, which processes the index data and computes the daily GMI score, ultimately determining if the current market regime is "Green" or "Red."
4. **Decision Point - GMI Status Check:** The system evaluates the GMI output.
5. **Path A (GMI is "Red"):** If the market is deemed unfavorable, the system logs this status, bypasses all individual stock screening, and terminates the process.
6. **Path B (GMI is "Green"):** If the market is favorable, the system proceeds to the next stage.
7. **Load Stock Universe:** The application identifies the list of ticker symbols to be screened (e.g., by scanning a directory of CSV files).
8. **Initiate Screening Loop:** The system begins to iterate through each ticker in the universe.
   - ○ **Load Ticker Data:** For the current ticker, its historical OHLCV data is loaded into a pandas DataFrame.
   - ○ **Apply Signal Algorithms:** The functions for the Green Line Breakout, Blue Dot, and Black Dot screeners are executed on the ticker's data.
   - ○ **Collect Signals:** If any of the algorithms identify a valid signal for the most recent trading day, the ticker, signal type, and relevant data (e.g., closing price) are stored in a results list.
9. **End Screening Loop:** Once all tickers have been processed, the loop concludes.
10. **Generate Final Report:** The collected signals are formatted into a clean, human-readable report (e.g., a CSV file).
11. **End Execution:** The program terminates.

The Market Regime Filter: Implementing the General Market Index (GMI)

The GMI serves as the system's foundational component, acting as a macro-level filter to assess the health of the overall market. Its purpose is to align trading activity with the "line of least resistance," a concept championed by market legends like Jesse Livermore and central to Dr. Wish's approach.[2] While the precise proprietary formula for the GMI is not publicly available, its signaling mechanism is clearly defined, providing a solid basis for implementation. This section details the known signal logic and proposes a robust proxy model to bridge the calculation gap.

Deconstructing the GMI Signal Logic

Analysis of Dr. Wish's presentation materials reveals a clear, rule-based trigger for the GMI's bullish and bearish signals.[2] The model is not a simple binary switch but incorporates a two-day confirmation period, a common technique designed to filter out short-term noise and reduce the likelihood of "whipsaw" signals at market turning points.

The explicit rules are as follows:

- **GMI Green Signal (Bullish Market):** A "GMI Green" signal is triggered when the underlying GMI numerical value is greater than 3 for two consecutive days. This indicates that conditions are favorable for taking new long positions.
- **GMI Red Signal (Bearish Market):** A "GMI Red" signal is triggered when the GMI numerical value is less than 3 for two consecutive days. This indicates that market conditions are unfavorable, and a defensive posture (e.g., holding cash, avoiding new long positions) is warranted.

This logic implies that the GMI is a composite indicator that produces a daily numerical score, which oscillates above and below a neutral threshold of 3. The core challenge is to reconstruct the methodology for calculating this daily score.

Addressing the Calculation Gap: Designing a GMI Proxy Model

Given that the specific components and weighting of the GMI formula are not disclosed in the available research [2], a GMI Proxy model (

GMI-P) must be constructed. This proxy will not be an exact replication but an engineered approximation designed to capture the analytical spirit of Dr. Wish's market assessment. The construction of this proxy is guided by the tools and concepts he repeatedly emphasizes for gauging market direction: trend analysis using moving averages, market breadth indicators, and principles aligned with the CANSLIM methodology.[1]

The proposed GMI-P will be a composite score calculated daily, ranging from 0 to 6. The score is derived by evaluating six distinct conditions. For each condition that is met on a given day, one point is added to the daily GMI-P score. The NASDAQ 100 ETF (QQQ) is used as the primary instrument for trend analysis, reflecting a focus on the growth-oriented segment of the market.

The six components of the GMI-P are:

1. **Short-Term Trend Confirmation:** The closing price of the QQQ must be above its 50-day simple moving average (SMA). The 50-day SMA is a widely followed proxy for the intermediate-term trend (equivalent to a 10-week moving average).
2. **Short-Term Trend Momentum:** The value of the 50-day SMA must be greater than its value on the previous day. This condition ensures the trend is not merely flat but is actively rising, confirming upward momentum.
3. **Long-Term Trend Confirmation:** The closing price of the QQQ must be above its 150-day SMA. This serves as a check on the longer-term health of the market, aligning with Dr. Wish's use of weekly charts (the 150-day SMA is a proxy for the 30-week SMA) for clearer trend identification.[5]
4. **Market Breadth:** The T2108 indicator, which measures the percentage of stocks trading above their 40-day moving average, must be above 50%. Dr. Wish frequently references T2108 as a key tool for assessing the underlying strength and participation in a market advance.[4] A reading above 50% suggests broad market participation.
5. **Short-Term Price Momentum:** The QQQ must have closed higher on at least two of the preceding three trading days. This is a simple measure to confirm the presence of immediate buying pressure.
6. **Absence of Institutional Selling:** There must not have been any "distribution days" on the QQQ within the last 10 trading sessions. A distribution day is defined here as a day where the index closes down by more than 1% on volume that is higher than the previous day's volume. This concept is a cornerstone of the CANSLIM system, designed to detect subtle signs of institutional selling pressure.[2]

The following table provides a clear, at-a-glance reference for the construction of the GMI Proxy model. This transparent documentation of the logic and its sources allows for understanding, validation, and potential future modification of the model.

Component ID	Description	Condition for +1 Point	Rationale / Source Reference
GMI-P1	Short-Term Trend	QQQ.Close > SMA(QQQ.Close, 50)	Foundational trend-following principle using key moving averages for market direction assessment.[3]
GMI-P2	Short-Term Trend Momentum	SMA(QQQ.Close, 50) > SMA(QQQ.Close, 50)[-1]	Confirms the trend is actively rising, not just consolidating above the average.
GMI-P3	Long-Term Trend	QQQ.Close > SMA(QQQ.Close, 150)	Based on the use of longer-term weekly

			charts (30-week MA) for superior trend clarity.[5]
GMI-P4	Market Breadth	T2108 > 50	Direct inclusion of a key breadth indicator frequently used by Dr. Wish for market timing.[4]
GMI-P5	Short-Term Price Momentum	Count(QQQ.Close > QQQ.Close[-1], 3) >= 2	Simple measure to ensure that short-term buying pressure is currently active in the market.
GMI-P6	Absence of Institutional Selling	No distribution days in the last 10 sessions	Aligns with CANSLIM principles of monitoring for institutional distribution, a key market health indicator.[2]

The Breakout Engine: Implementing the Green Line Breakout (GLB) Screener

The Green Line Breakout (GLB) is Dr. Wish's signature strategy for identifying stocks with the potential for explosive upward moves. The setup is designed to capture the moment a stock emerges from a long period of consolidation to new, unexplored price territory.[5] The core psychology behind the GLB is that a stock breaking out to a new all-time high has no "overhead resistance"—there are no prior buyers holding at a loss who are waiting to sell at their breakeven point.[6] This situation, especially when accompanied by strong institutional buying (evidenced by high volume), is considered one of the most bullish technical events.[3] This section provides a precise, step-by-step algorithm for identifying GLB signals, synthesizing the descriptive rules from multiple sources with the implementation logic observed in public code examples.

Defining the GLB Criteria

The rules for a valid GLB are consistent across Dr. Wish's presentations and related educational materials.[6] A successful implementation must adhere strictly to these criteria:

1. **All-Time High (ATH):** The stock must establish a new all-time high price. This is the starting point for the pattern.
2. **Consolidation Period:** Following the establishment of this ATH, the stock must undergo a period of consolidation or "rest" for a minimum of three full calendar months. During this period, the monthly high of the stock must not exceed the price of the ATH. This period of sideways movement is interpreted as a phase of accumulation by institutional investors.[6]
3. **The "Green Line":** The price level of the ATH that precedes the three-month consolidation becomes the "Green Line." This horizontal line on the chart represents a critical resistance level.
4. **The Breakout:** A valid GLB signal occurs when the stock's daily closing price moves decisively above the Green Line. This action signifies that the accumulation phase is complete and a new markup phase is beginning.
5. **Volume Confirmation:** The breakout day should be accompanied by a significant surge in trading volume, ideally at least 50% above the 50-day average volume. This high volume confirms institutional participation and validates the authenticity of the move.[6]
6. **Optional Strength Filter:** As an additional filter for identifying the most powerful candidates, Dr. Wish notes a preference for stocks that have already doubled in price from their 52-week low *before* establishing the Green Line top.[7] While not a strict requirement, this characteristic suggests powerful underlying momentum.

Algorithmic Implementation Steps

The implementation of the GLB screener requires a multi-timeframe analysis approach. The structural pattern (the Green Line itself) is defined using monthly data to filter out short-term noise, while the entry trigger (the breakout) is a daily event.[9] This separation of long-term structure from short-term action is a hallmark of sophisticated technical analysis and is crucial for a correct implementation.

The algorithm proceeds as follows for each individual stock:

- **Step 1: Resample to Monthly Data.**
  - Load the complete daily OHLCV history for the stock into a pandas DataFrame.
  - Resample this daily data into monthly data. The key value needed is the maximum High for each calendar month. This can be achieved in pandas using Monthly_Highs = Daily_DataFrame['High'].resample('M').max().
- **Step 2: Identify the Most Recent Valid Green Line.**
  - This is the most complex part of the algorithm. The logic must iterate backward

through the monthly data to find a high that meets the consolidation criteria.

- Start from the second-to-last completed month (M-1) and iterate backward.
- For each month i in the history, let potential_GL_price = Monthly_Highs[i].
- Check if this potential_GL_price is the highest high in all preceding months (Monthly_Highs[j] < potential_GL_price for all j < i). This confirms it was an ATH at that time.
- If it was an ATH, check the consolidation condition: Monthly_Highs[i+1] < potential_GL_price, Monthly_Highs[i+2] < potential_GL_price, and Monthly_Highs[i+3] < potential_GL_price.
- If both the ATH and consolidation conditions are met, a valid Green Line has been found at potential_GL_price. Store this value and the date it was established. Since we are looking for the most recent setup, the first one found when iterating backward will be the correct one.

- **Step 3: Detect the Daily Breakout Event.**
    - Using the daily DataFrame, check the two most recent trading days.
    - A breakout signal is confirmed if:
        - Daily_DataFrame['Close'].iloc[-1] > potential_GL_price (Today's close is above the Green Line).
        - Daily_DataFrame['Close'].iloc[-2] <= potential_GL_price (Yesterday's close was at or below the Green Line).
    - This two-part condition ensures that the signal is triggered only on the exact day of the breakout.
- **Step 4: Apply Volume Confirmation Filter.**
    - If a daily breakout is detected, calculate the 50-day simple moving average of volume (SMA_Volume_50).
    - Check if the volume on the breakout day meets the confirmation criteria: Daily_DataFrame['Volume'].iloc[-1] > (SMA_Volume_50.iloc[-1] * 1.5). The multiplier (1.5) should be a configurable parameter.
- **Step 5: Apply Optional "Doubled" Filter.**
    - If a valid, volume-confirmed breakout has occurred, perform this final check.
    - Identify the date the Green Line was established (from Step 2).
    - Look back 252 trading days (approximately one year) from that date and find the minimum Low price in that period (Low_52_Week).
    - Check if potential_GL_price >= (Low_52_Week * 2).
    - The result of this check (True/False) should be included in the final output report to allow the user to prioritize these stronger candidates.

The Pullback Engine: Implementing Blue Dot and Black Dot Signals

In addition to momentum breakouts, Dr. Wish's methodology includes setups for buying pullbacks within established uptrends. These are identified by the "Blue Dot" and "Black Dot" signals. Both are based on the Stochastic Oscillator, a momentum indicator that measures a security's closing price relative to its high-low range over a set period.[12] These signals are designed to identify low-risk entry points where a stock is temporarily "oversold" and likely to resume its primary uptrend.[1] While similar in purpose, the two signals have distinct rules and parameters, reflecting different levels of strictness in their filtering criteria.

The Blue Dot Algorithm: Oversold Bounce in a Confirmed Uptrend

The Blue Dot is a high-conviction signal designed to pinpoint an oversold bounce in a stock that is in a strong, clearly defined uptrend.[3] The logic is precise and combines a specific stochastic event with a strict trend confirmation filter. The algorithm, derived directly from a public Pine Script implementation, ensures a faithful replication of the signal's intent.[14]
A Blue Dot signal is generated on a given day if and only if **all three** of the following conditions are met:

1. **Stochastic Oversold State (Previous Day):** The value of the 10-period Stochastic %K line on the *previous* trading day must have been less than 20. This establishes that the stock was in an oversold condition.
   - Calculation: Stochastic %K(10)[-1] < 20
2. **Stochastic Bullish Crossover (Current Day):** The value of the 10-period Stochastic %K line on the *current* trading day must be greater than 20. This, combined with the first condition, defines the precise crossover event, signaling that short-term momentum is turning up from an oversold level.
   - Calculation: Stochastic %K(10) > 20
3. **Uptrend Confirmation (Rising Moving Average):** The value of the 50-period simple moving average of the closing price on the current day must be greater than its value on the previous day. This is a strict trend filter, requiring not only that the price is in an uptrend but that the trend itself is actively accelerating or maintaining its upward slope.
   - Calculation: SMA(Close, 50) > SMA(Close, 50)[-1]

The Black Dot Algorithm: A More Permissive Stochastic Bounce

The Black Dot signal serves a similar purpose to the Blue Dot but employs more lenient criteria, allowing it to trigger in a wider variety of market conditions.[1] It is designed to catch potential swing lows even if the uptrend is less mature or choppier. The algorithm is derived from a detailed analysis of a ThinkScript implementation shared within trading communities.[15]

A Black Dot signal is generated on a given day if and only if **all three** of the following conditions are met:

1. **Stochastic Oversold Window:** The value of the Full Stochastic Oscillator (with parameters 10 for the %K period and 1 for the %D period) must have been less than or equal to 25 on *any one* of the three preceding trading days. This creates a three-day lookback window for the oversold condition, making it less dependent on a single day's reading.
   ○ Calculation: Sum(StochasticFull(10,1) <= 25, 3) >= 1
2. **Immediate Upward Momentum:** The closing price on the current day must be greater than the closing price on the previous day. This is a simple but effective confirmation that buying pressure has returned in the immediate short term.
   ○ Calculation: Close > Close[-1]
3. **Permissive Trend Confirmation:** The closing price on the current day must be above *either* its 30-day simple moving average *or* its 21-day exponential moving average. This trend filter is significantly more permissive than the Blue Dot's. A stock can meet this condition even if the moving averages themselves are flat or slightly declining, as long as the price is trading above them.
   ○ Calculation: (Close > SMA(Close, 30)) OR (Close > EMA(Close, 21))

The differences between these two signals highlight a key trade-off in strategy design. The Blue Dot's stringent requirements—a precise stochastic crossover and a rising 50-day SMA—filter for "A+" quality setups in strong, stable uptrends. This will likely result in fewer signals but a higher win rate. Conversely, the Black Dot's more flexible criteria—a 3-day oversold window and a less restrictive "price above MA" trend filter—will generate more signals. This increased frequency comes at the probable cost of a lower win rate, as it will identify setups in less ideal trend conditions. A comparative backtest would be essential to quantify this performance trade-off.

The following table provides a side-by-side comparison to clearly delineate the implementation logic for each signal, preventing confusion and ensuring accurate coding.

Parameter	Blue Dot Signal	Black Dot Signal
Stochastic Type	Standard %K	Full %K
Stochastic Period	10	10, 1
Oversold Condition	K[-1] < 20 AND K > 20 (Strict Crossover)	K <= 25 in any of the last 3 days (Lenient Window)
Momentum Trigger	Implicit in the stochastic crossover	Close > Close[-1] (Explicit Price Action)
Trend Confirmation	SMA(50) > SMA(50)[-1] (Strict, Rising MA)	Close > SMA(30) OR Close > EMA(21) (Permissive, Above MA)
Source Logic	Pine Script Analysis [14]	ThinkScript Forum Analysis [15]

Integration, Screening, and Operational Workflow

With the individual algorithmic components for market timing (GMI) and signal generation (GLB, Blue Dot, Black Dot) defined, this section outlines the practical assembly of these modules into a cohesive, automated daily screening application. The focus is on creating a modular, maintainable, and efficient system that executes the hierarchical workflow and produces a clear, actionable output.

The Main Execution Script (main.py)

The core of the application will be a single orchestrator script, main.py. This script will not contain the complex algorithmic logic itself but will be responsible for managing the overall workflow, handling configuration, and calling the specialized functions from other modules.
The primary responsibilities of main.py will be:

- **Configuration Management:** At the start of the script, it will define key configuration variables, such as the file path to the historical data directory, the ticker symbol for the market index (e.g., 'QQQ'), and the name of the output file.
- **Workflow Orchestration:** It will execute the screening process in the precise sequence defined in Section 1.3:
    1. Call the GMI function to determine the market state.
    2. Based on the GMI state, either terminate or proceed to individual stock screening.
    3. If proceeding, it will manage the loop that iterates through all available stock tickers.
    4. Within the loop, it will call the respective screener functions for GLB, Blue Dot, and Black Dot.
    5. It will collect all positive signals into a master list.
    6. Finally, it will call the reporting function to save the results to a file.

Modular Code Structure

To promote code clarity, reusability, and ease of maintenance, the project will be organized into a series of distinct Python modules (files), each with a specific responsibility. This separation of concerns is a fundamental principle of good software design.
The proposed file structure is as follows:

- main.py: The main entry point and orchestrator of the application, as described above.
- data_loader.py: A module containing functions responsible for loading and pre-

processing the OHLCV data from CSV files into pandas DataFrames. This centralizes all data input operations.
- gmi_calculator.py: This module will contain the function(s) necessary to calculate the daily GMI-P score and return the final market state ("Green" or "Red"). It will take the market index DataFrame as input.
- glb_screener.py: This module will house the function that implements the Green Line Breakout algorithm. It will take an individual stock's DataFrame as input and return a boolean value indicating if a GLB signal is present, along with the Green Line price level if applicable.
- dot_screeners.py: This module will contain two separate functions: one for the Blue Dot algorithm and one for the Black Dot algorithm. Each function will take a stock's DataFrame as input and return a boolean signal.
- reporting.py: This module will contain a function that takes the final list of collected signals and formats it into a pandas DataFrame, which is then saved to a CSV file with a timestamped filename.

Final Output Generation

The primary artifact produced by the daily run of the application will be a simple, clean, and data-rich CSV file named according to the date of the screening (e.g., signals_2024-10-26.csv). This format is easily readable by both humans and other software (like spreadsheet programs) for further analysis.

The output file will contain the following columns:

- Date: The date on which the signal was generated.
- Ticker: The stock symbol that triggered the signal.
- Signal_Type: The name of the specific setup (e.g., "GLB", "Blue Dot", "Black Dot").
- Close_Price: The closing price of the stock on the signal day.
- Volume: The trading volume on the signal day.
- GLB_Details: For GLB signals, this column will contain the price of the Green Line that was breached. For other signal types, this field will be empty. This provides crucial context for evaluating the breakout's magnitude.

The following table outlines the key Python libraries and their recommended versions, providing an actionable checklist for setting up the project's development environment.

Library	Version (Recommended)	Primary Purpose
pandas	2.x or later	Core data manipulation, time-series analysis, DataFrame management.
numpy	1.2x or later	High-performance numerical computing and array operations.

ta or talib-wrapper	Latest	Accurate and efficient calculation of technical indicators (SMA, EMA, Stochastic Oscillator).
yfinance	Latest	(Optional) For fetching supplementary data, such as the T2108 index, if not available locally.

Advanced Considerations and Path Forward

Developing the screener as described in the preceding sections will produce a powerful tool for identifying trade setups according to Dr. Wish's methodologies. However, moving from a signal generation tool to a comprehensive and robust trading system requires addressing several practical challenges and considering future enhancements. This final section provides expert guidance on signal validation, performance measurement, and potential avenues for expanding the system's capabilities.

Managing Signal Quality and False Positives

No technical signal is infallible. Breakouts, in particular, are susceptible to "false head fakes," where a stock briefly pierces a resistance level only to reverse lower, trapping eager buyers.[16] Similarly, pullback signals like the Blue and Black Dots can fail if the underlying trend weakens or the broader market sentiment shifts negatively.[3] To improve the robustness of the generated signals, several additional filtering layers can be implemented.

- **GLB Confirmation Rules:** To mitigate the risk of failed breakouts, post-signal confirmation criteria can be added. Instead of acting on the first close above the Green Line, the system could be configured to require:
  - A **two-day close confirmation**, where the stock must close above the Green Line for two consecutive days before the signal is considered valid.
  - A **pullback and retest confirmation**, where the system waits for the initial breakout and then watches for the price to return to the Green Line level and successfully "bounce" off it, confirming its transition from resistance to support.[6]
- **Relative Strength (RS) Filter for Dot Signals:** To ensure that Blue and Black Dot signals are only considered in market-leading stocks, a Relative Strength filter can be applied. This concept, central to the CANSLIM methodology that Dr. Wish respects, compares a stock's price performance to that of the broader market (e.g., the S&P 500).[2] A common implementation is to calculate an RS Rating (from 1 to 99) and only

accept Dot signals in stocks with a rating above a certain threshold (e.g., 80). This ensures that the system is buying pullbacks in stocks that are already outperforming the market, increasing the probability of a swift trend resumption.

Backtesting and Performance Validation

A screener identifies setups; it does not, by itself, validate a strategy's profitability. The crucial next step is to conduct a rigorous historical backtest to evaluate the performance of the signals generated by the system. This involves creating a backtesting engine that can simulate trading the signals over a long historical period.

A basic backtesting framework would require:

1. **Defining Entry and Exit Rules:** For each signal type, define unambiguous rules for trade management. For example:
    - **Entry:** Buy at the closing price on the day the signal is confirmed.
    - **Initial Stop-Loss:** Place an initial stop-loss at a predefined level, such as 7-8% below the entry price or below the low of the signal day's candle.
    - **Profit Target / Trailing Stop:** Define rules for taking profits, such as selling after a 20-25% gain or using a trailing stop based on a moving average (e.g., the 21-day EMA).[6]
2. **Calculating Performance Metrics:** After running the simulation, calculate key performance metrics to objectively assess the strategy, including:
    - **Total Return:** The overall percentage gain or loss.
    - **Win Rate:** The percentage of trades that were profitable.
    - **Profit Factor:** Gross profits divided by gross losses.
    - **Maximum Drawdown:** The largest peak-to-trough decline in equity.
    - **Sharpe Ratio:** A measure of risk-adjusted return.
3. **Comparative Analysis:** The backtest should be run separately for each signal type (GLB, Blue Dot, Black Dot) to compare their performance characteristics and identify which setups are most effective.

Parameter Optimization and Sensitivity Analysis

The system relies on several fixed parameters or "magic numbers," such as the 3-month consolidation period for a GLB, the 10-period lookback for stochastics, or the 50-day period for the moving average trend filter. While these are the documented values, it is prudent to understand how sensitive the system's performance is to variations in these parameters. A sensitivity analysis involves running multiple backtests while systematically altering one parameter at a time to observe the impact on performance. This helps to determine if the strategy's success is robust or if it depends precariously on a specific, optimized value. This process is critical for avoiding "overfitting," where a system is tuned so perfectly to historical data that it fails to perform in live market conditions.

Future Enhancements

Once the core screening and backtesting functionality is established, several enhancements can be added to increase the system's power and utility:

- **Integration of Fundamental Filters:** Dr. Wish's approach is rooted in the CANSLIM tradition, which combines strong technicals with strong fundamentals.[2] A significant enhancement would be to integrate a fundamental data source and add filters to the screener. For example, the system could be configured to only show GLB signals for companies that also exhibit strong quarterly earnings per share (EPS) and sales growth (e.g., >25%).
- **Automated Alerting:** The final script can be integrated with messaging services (e.g., via email using smtplib or services like Telegram or Slack via their APIs). This would allow the system to run automatically on a server each day after the market close and send real-time alerts of new signals directly to a mobile device or desktop.
- **Automated Chart Generation:** To facilitate quick visual verification of signals, the system could be enhanced to use a plotting library like matplotlib or mplfinance. Upon detecting a signal, it could automatically generate and save a chart of the stock, with the relevant indicators and patterns (e.g., the Green Line drawn horizontally, Blue/Black Dots plotted below the price bars) clearly annotated. This would save significant time during the daily review process.

Works cited

1. Lessons from 50 Years of Trading Part-Time | Dr. Eric Wish - YouTube, accessed August 18, 2025, https://www.youtube.com/watch?v=WvFxOyMz5z0
2. Eric D. Wish, Ph.D., accessed August 18, 2025, https://aaiirtp.org/wp-content/uploads/2021/12/Dr.Wish2021-12-11.pdf
3. The Power of Setups: Dr. Eric Wish's Guide to Finding, Timing & Executing Winning Trades, accessed August 18, 2025, https://www.youtube.com/watch?v=dl7lGdHuX2U
4. How to Time The Stock Market and Determine the Trend | 50 Year ..., accessed August 18, 2025, https://www.youtube.com/watch?v=AH7YXJrpRc4
5. The Green Line Breakout Setup and Insights from 50 Years of Trading (Must Watch!), accessed August 18, 2025, https://www.youtube.com/watch?v=fYU-cz9j61g
6. Trading The Green Line Breakout: 6 Steps For Breakout Trades - TraderLion, accessed August 18, 2025, https://traderlion.com/technical-analysis/green-line-breakout/

7. Green Line Breakout (GLB) Explained GMI Remains Green | PDF | Stocks - Scribd, accessed August 18, 2025, https://www.scribd.com/document/439682087/Green-Line-Breakout-GLB-Explained-GMI-Remains-Green
8. Dr. Eric Wish's Green Line (GLB) — Indicator by DuckSoupForex - TradingView, accessed August 18, 2025, https://www.tradingview.com/script/0mUjAA6d-Dr-Eric-Wish-s-Green-Line-GLB/
9. Green Line Breakout by Eric Wish - The StockCharts Answer Network (s.c.a.n.), accessed August 18, 2025, https://scan.stockcharts.com/discussion/2477/green-line-breakout-by-eric-wish
10. Green Line Breakout (GLB) — Indicator by GreatStockMark - TradingView, accessed August 18, 2025, https://www.tradingview.com/script/d6hg0wPr-Green-Line-Breakout-GLB/
11. Green Line Breakout (GLB) - Public Use — Indicator by ..., accessed August 18, 2025, https://www.tradingview.com/script/qfNJJ8Da-Green-Line-Breakout-GLB-Public-Use/
12. Stochastic Oscillator (Fast, Slow, and Full) - ChartSchool - StockCharts.com, accessed August 18, 2025, https://chartschool.stockcharts.com/table-of-contents/technical-indicators-and-overlays/technical-indicators/stochastic-oscillator-fast-slow-and-full
13. The Green Dot Indicator - YouTube, accessed August 18, 2025, https://www.youtube.com/watch?v=TGyreX-m5L0
14. Blue Dot — Indicator by wiz_trades_ — TradingView, accessed August 18, 2025, https://www.tradingview.com/script/Daql0Dxr-Blue-Dot/
15. Dr Wish's Black & Green Dot Indicator For ThinkOrSwim ... - thinkScript, accessed August 18, 2025, https://usethinkscript.com/threads/dr-wishs-black-green-dot-indicator-for-thinkorswim.8110/
16. How To Profit From the Bollinger Squeeze - Investopedia, accessed August 18, 2025, https://www.investopedia.com/articles/technical/04/030304.asp
17. What am i doing wrong? keep falling for false breakouts. : r/Daytrading - Reddit, accessed August 18, 2025, https://www.reddit.com/r/Daytrading/comments/1h7rdc4/what_am_i_doing_wrong_keep_falling_for_false/