[Горячее](#) [Лучшее](#) [Свежее](#) [Подписки](#) [empenoso](#) 8 дней назад  Программирование на python

Я посадил торгового робота в стену. Он теперь за меня торгует

В предыдущих статьях я рассказывал, как пришёл к идее создания собственного торгового робота. Мотивация проста:

- Автоматизация - алгоритм не спит, не нервничает и не занят своими делами.
- Дисциплина - робот исключает эмоции, следуя правилам.
- Тестирование - любую идею можно проверить на исторических данных, прежде чем рисковать деньгами.

Я всегда разделял два этапа: разработку торговых идей (логика стратегии) и реализацию механизма исполнения (отправка заявок, автотрейдинг). Сначала - бэктестинг и базовая оптимизация, и только потом - реальная торговля.

Поскольку я нахожусь в активном поиске подходящего решения для автотрейдинга и уже опробовал несколько рабочих вариантов, то эта статья представляет мои размышления об этом механизме исполнения заявок. **Ваша критика или поддержка идей приветствуется.**

Почему я не хочу использовать QUIK и Windows?

По моему мнению QUIK архаичен, нестабилен для автоматизации и требует оконной среды. Он не предназначен для headless-серверов (это компьютер без монитора, клавиатуры, мыши). QUIK + LUA или внешнее ПО - это сложная, криво документированная и уязвимая связка.

Выбор железа и ОС: Raspberry Pi + Ubuntu Server

Используйте аккаунт Яндекса для входа на сервис

Безопасный вход без дополнительной регистрации на сайте

[Войти с Яндекс ID](#)[Войти](#)[Создать аккаунт](#)[Забыли пароль?](#)

или продолжите с

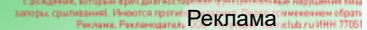
[Войти с Яндекс ID](#)[Войти через VK ID](#)[Промокоды](#)[Работа](#)[Курсы](#)[Реклама](#)[Игры](#)[Пополнение Steam](#)

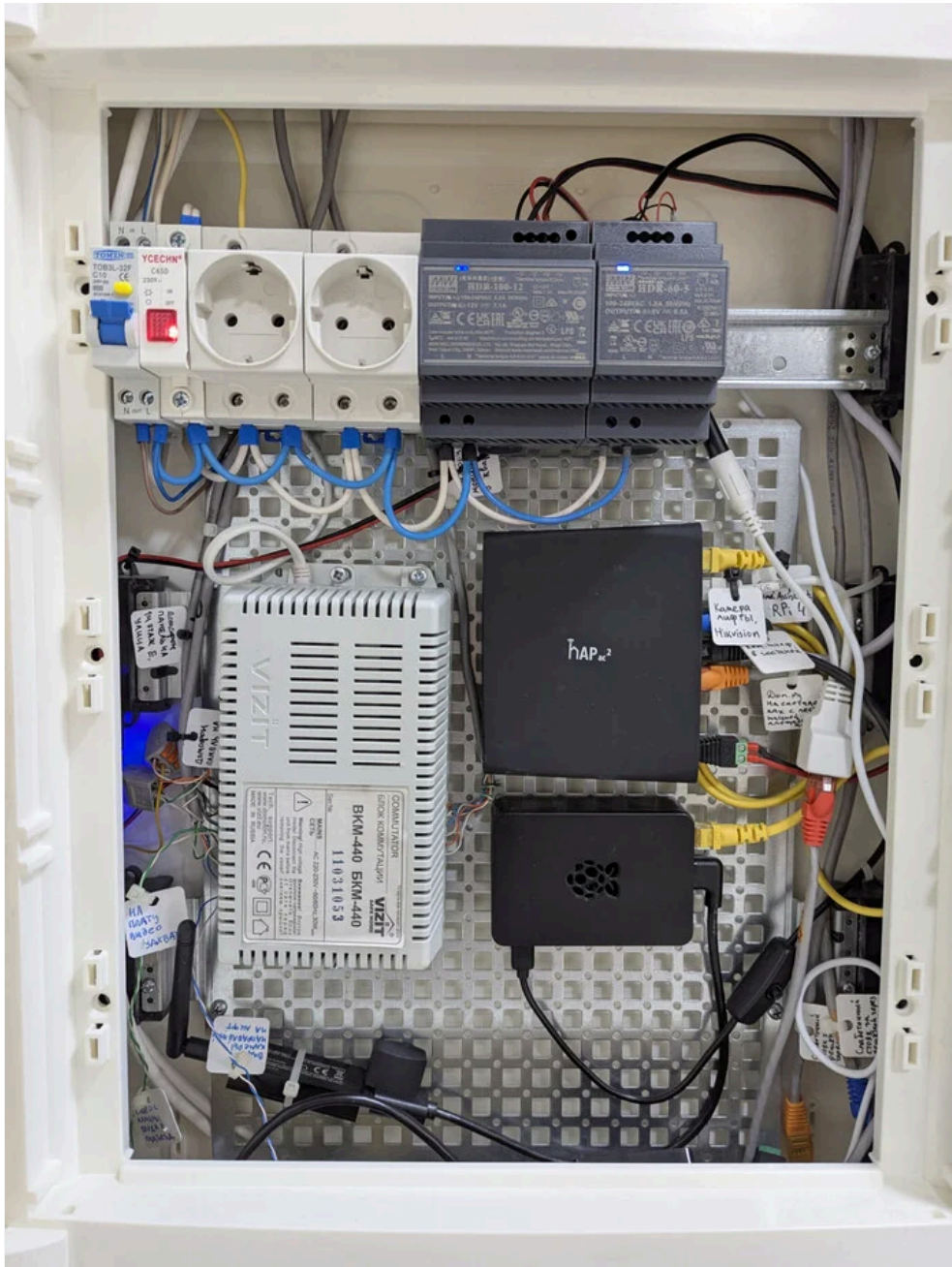


недорогой если брать не самые новые модели.



[Посмотреть весь топ](#)





Слаботочный шкаф. Raspberry Pi 4 в коробке со стилизованным изображением малины (raspberrу). На этой малинке на фото крутится Home Assistant для управления всеми умными устройствами

В качестве операционной системы я выбрал Ubuntu Server. Этот дистрибутив Linux славится стабильностью, обширным сообществом и легкостью установки нужного ПО. Python, библиотеки для работы с API - всё ставится без проблем. Отсутствие графического интерфейса (GUI) экономит драгоценные ресурсы "малинки", а управление системой удобно осуществляется удаленно по SSH с любого другого компьютера в домашней сети.

Конечно, можно было бы рассмотреть аренду виртуального сервера (VPS) в облаке, но это влечет ежемесячные платежи. Старый ноутбук тоже мог бы подойти, но Raspberry Pi компактнее, тише и может быть установлена незаметно. Поэтому для моих целей связка Raspberry Pi и Ubuntu Server выглядит оптимальным решением для создания независимого и надежного торгового бота.



Лучшие посты
недели



Рассылка Пикабу:
отправляем самые
рейтинговые материалы за 7
дней 🔥

Укажите

Подписаться

Нажимая кнопку
«Подписаться на рассылку»,
я соглашаюсь с [Правилами](#)
[Пикабу](#) и даю согласие на
обработку персональных
данных.

MEDIASNIPIR

ОБЛЕГЧАЕТ СИМПТОМЫ АЛЛЕРГИИ
БЫСТРО



за 7 д

Реклама. ООО «

Симптомы аллергии к белку коровьего молока у детей до 1 года. 7 дней
упрощенной симптоматики по результатам исследования И.Б. Макаровой и
применения смеси на основе гидролизата сывороточного белка «Ступа»
в диетотерапии детей с аллергией к белку коровьего молока. Гидролизат, 1
Информация для родителей детей с аллергией к белку коровьего молока «Аллергия»
молочка. Дети имеют с рождения до 1 года жизни. 7 дней
облегчения симптоматики. 1 год жизни. 7 дней

Реклама

Новости Пикабу
Помощь Награды
Кодекс Пикабу
Реклама Контакты
О компании О проекте
Зал славы

Промокоды Купоны Мегамаркет
Скидки Купоны AliExpress
Работа Купоны M.Видео
Курсы Купоны YandexTravel
Блоги Купоны Lamoda

Мобильное приложение



Микрокомпьютеров существует довольно таки много: например этот старый Orange Pi у меня служит для превращения обычного квартирного домофона в IP камеру видеонаблюдения [Как превратить обычный квартирный домофон в IP камеру видеонаблюдения](#)

Поиск подходящего языка и фреймворка

Вообще начал я с того, что было ближе и привычнее - Node.js. У меня за плечами немалый опыт в этом языке. Работа с WebSocket и REST API, запуск задач по таймеру, хранение данных в JSON - всё это в Node делается быстро и удобно. Именно на этой технологии я собрал свой первый рабочий прототип, и даже выложил его в открытый доступ: [SilverFir-TradingBot](#).

Но по мере развития проекта я начал сталкиваться с ограничениями. Самая большая проблема - отсутствие зрелых библиотек для финансового анализа и бэктестинга. Всё приходилось писать вручную: от парсинга котировок до расчёта индикаторов. Подключение к API брокеров тоже часто требовало обёрток и промежуточных слоёв.

Под капотом Pine Script: как устроен и для чего используется язык TradingView

В процессе изучения темы я также узнал о Pine Script - языке программирования, встроенном в TradingView. Он специально создан для написания торговых индикаторов и стратегий. Прост в освоении, отлично подходит для визуального анализа на графике, позволяет запускать простейшие бэктесты буквально в пару строк. Но у него есть ограничения: нет доступа к внешним API и он годится скорее для прототипирования идей, чем для полноценного автотрейдинга.

Поэтому я переключился на Python - язык, на котором написано большинство современных библиотек для финансов, анализа данных и машинного обучения. С переходом на Python открылся целый мир: Backtrader, pandas, matplotlib, TA-Lib. Всё

готово: бери и тестируй любую идею. Вместо бесконечных "велосипедов" я сосредоточился на том, что важно - стратегии.

От Backtrader к Pine Script и обратно

Сначала я выбрал **Backtrader** - мощный инструмент, позволяющий детально протестировать стратегию, использовать несколько таймфреймов, фильтры, индикаторы, комиссионные и проскальзывание. Но очень быстро стало ясно, что у него есть обратная сторона: слишком много времени уходит на подготовку данных, настройку окружения, отладку визуализации и описание каждой мелочи в коде. Всё гибко, но громоздко.

Мой первый и неудачный опыт поиска торговой стратегии для Московской биржи

Решил попробовать **backtesting.py** - тем более эта библиотека недавно обновилась. Она действительно проще, особенно если тестировать одну стратегию на одной бумаге. Но всё равно: слишком много ручной настройки, особенно если часто меняешь параметры.

Тестировании торговой системы со случайными сигналами на вход для фьючерсов Московской биржи при помощи Python

В какой-то момент я открыл для себя **Pine Script** и **TradingView**. Это было откровением. Пишешь код - сразу видишь результат. Делиться стратегией можно по ссылке, и любой читатель статьи (я ведь пишу публичные статьи) мгновенно видит то же, что и я. Идеально для демонстрации логики и быстрого прототипирования. Но автоматизировать торговлю через Pine Script - плохая идея: нет доступа к API, нет гарантий исполнения.

Pine Script в деле: тестируем стратегию с линейной регрессией и R^2 (по мотивам S&C из 2007 года) на Московской Бирже

Я начал искать альтернативы и наткнулся на **QuantConnect (Lean Engine)**. Вроде бы всё есть — история, брокеры, мультиассеты. Но реализация на C#, закрытая инфраструктура и сложность кастомизации меня остановили.

В итоге я хочу вернуться к Python и Backtrader. Хотя он мне и не особенно нравится, но в связке с готовой библиотекой-оберткой он позволяет работать с реальным API. Это единственный реально доступный путь автоматизации с брокером в России, не завязанный на QUIK и Windows.

Место установки сервера в стене, где стоит коммутатор и один из роутеров

Обзор публичных API российских брокеров

В России есть выбор. Сегодня три брокера:

- Т-Инвестиции: <https://developer.tbank.ru/invest/intro/intro>
- Алор: <https://alor.dev/docs/>
- Финам: <https://trade-api.finam.ru/swagger/index.html>

предоставляют официально задокументированные API.

Для всех них энтузиаст из Екатеринбурга Чечет Игорь Александрович разработал удобные Python-обёртки: [TinkoffPy](#), [AlorPy](#) и [FinamPy](#). Это даёт возможность не только отправлять заявки, но и проводить бэктестинг прямо из Backtrader. Например, связка [BackTraderTinkoff](#) позволяет торговать и тестировать стратегии, не выходя из Python-экосистемы.

На фоне этого особенно интересно сравнить с международной практикой. У брокера Interactive Brokers [есть официальное руководство по работе с Backtrader](#) - такой уровень зрелости интеграции говорит о серьёзном подходе. Аналогично и в криптоиндустрии: тот же Bybit предоставляет API и готовые коннекторы на GitHub, что подчёркивает тренд на API-фикацию торговли. Я сам не работаю с криптой, но это внушает оптимизм.

Переделанное место установки Raspberry Pi в стене

Заключение

Итак, мой текущий стек вырисовывается так: Pine Script для быстрой проверки гипотез и визуализации, затем перенос логики в Python и Backtrader для глубокого бэктестинга, и наконец, TinkoffPy для реальной торговли через API брокера на домашнем Raspberry Pi.

Цель - полностью автономный робот на открытых технологиях, без QUIK и сопутствующей головной боли.

Что думаете?

Автор: Михаил Шардин

 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

20 мая 2025 г.



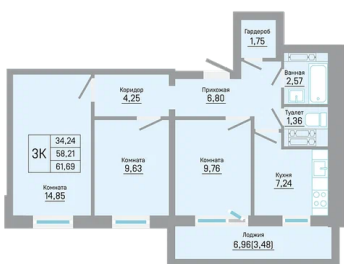
8 5.8K

Поддержать

Эмоции

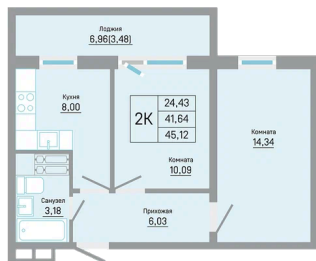
РЕКЛАМА

pzsp.ru



6 676 624 Р

3к квартира от ПЗСП. Старт продаж на Автозаводской!



4 491 012 Р

2к квартира от ПЗСП. Старт продаж на Автозаводской!



Программирование на python

850 постов • 11.9K подписчиков

Добавить пост

Подписаться



Правила сообщества

Публиковать могут пользователи с любым рейтингом. Однако!

Приветствуется:...

[Подробнее](#) ✓

Все комментарии Автора

Раскрыть 8 комментариев

Чтобы оставить комментарий, необходимо [зарегистрироваться](#) или [войти](#)



