

CryptoOracle Plus - 项目完全指南 & 常见问题 (FAQ)

本文档旨在记录项目的核心逻辑、配置建议以及针对不同市场环境的应对策略，涵盖了开发过程中遇到的关键问题与解决方案。

1. 核心交易逻辑

1.1 双频监控机制 (Dual-Frequency Monitoring)

为了解决 "大周期稳健" 与 "小周期敏捷" 之间的矛盾，我们采用了双频机制：

- 分析周期 (Timeframe):** 15m (推荐)。AI 基于 15 分钟的 K 线结构判断趋势，过滤短期噪音。
- 轮询间隔 (Loop Interval):** 15s (推荐)。机器人每 15 秒检查一次最新价格。如果价格发生剧烈波动，即使 K 线未走完，也能触发止损或捕捉机会。
- 配置方法:**

```
1 "timeframe": "15m",  
2 "loop_interval": 15
```

1.2 稳定币 vs 波动资产

AI 已内置特征识别，根据交易对自动切换策略：

- 稳定币对 (如 USDC/USDT):**
 - 策略:** 均值回归 (Mean Reversion)。
 - 逻辑:** 价格围绕 1.0000 波动。低于 0.9992 买入，高于 1.0008 卖出。忽略趋势指标。
- 波动资产 (如 BTC, MASK):**
 - 策略:** 趋势跟踪 + 结构分析。
 - 逻辑:** 关注 ADX (趋势强度) 和 K 线形态。

1.3 杠杆与风控感知

AI 在决策时会明确感知当前的 **杠杆倍数**：

- 如果杠杆为 10x，波动 1% = 盈亏 10%。
- AI 会据此收紧止损建议，防止高杠杆下的意外爆仓。

2. 关键配置详解

2.1 滑点保护 (Max Slippage)

- 参数:** max_slippage_percent
- 推荐值:** 2.0 (对于波动较大的山寨币/妖币)。
- 作用:** 允许成交价与分析价有 2% 的偏差。如果设得太小 (如 0.5%)，在暴涨行情中容易因为价格变动太快而买不进去 (踏空)。

2.2 资金分配 (Allocation)

- 参数: `allocation`
- 用法:
 - `0.1` = 使用账户 10% 的资金。
 - `1.0` = 使用账户 100% 的资金 (All-in)。
- 激进模式 (Aggressive Mode): 如果开启且 AI 信心为 `HIGH`, 机器人可能会突破单币种配额, 调用账户闲置资金 (上限 90%) 进行重仓。如果不希望这样, 请关闭 `enable_aggressive_mode`。

2.3 止盈止损 (Take Profit / Stop Loss)

这是**硬性防线**, 优先级高于 AI 建议。

- `max_profit_usdt`: 目标总盈利金额 (如 30 U)。一旦达到, 立即落袋为安。
- `max_loss_rate`: 强制止损比例 (如 0.05 即 5%)。一旦亏损触及, 无条件平仓。
- 注意: 现已将这两个目标传给 AI, AI 会努力配合您达成目标 (例如接近 30U 时主动建议止盈)。

3. 常见问题与解决方案 (Troubleshooting)

Q1: 浮盈波动太小, 怎么选币? (如何提升波动率)

如果您觉得收益 (或亏损) 波动太小, 可能是因为选到了低波动的“死鱼”币种。以下是几种解决方案:

方案 A: 优先选择 Meme 或热门板块

- **Meme 板块:** `DOGE`, `SHIB`, `PEPE`, `BONK`, `FLOKI`。这些币种日内波动经常超过 5%, 非常适合机器人的趋势捕捉策略。
- **热门公链/Layer2:** `SOL`, `SUI`, `OP`, `ARB`。流动性好且波动适中。
- **避开:** 老牌主流币 (如 `LTC`, `XRP`, `ADA`) 在非行情期波动极低, 容易横盘磨损手续费。

方案 B: 使用波动率扫描脚本 (Volatility Scanner)

我们在 `src/tools/` 下提供了一个扫描脚本, 可以帮您找出当前 OKX 上波动最大的币种。
(注: 如果您的项目目录下没有该脚本, 请参考下文手动创建)

1. 创建脚本 `scan_volatility.py`:

```
1  import ccxt
2  import pandas as pd
3
4  # 初始化
5  okx = ccxt.okx()
6  markets = okx.load_markets()
7
8  # 筛选 USDT 永续合约
9  symbols = [s for s in markets if s.endswith('/USDT:USDT')]
10
11 print(f"正在扫描 {len(symbols)} 个交易对的波动率...")
12 results = []
13
14 for symbol in symbols[:50]: # 演示只扫前50个, 全扫请去掉切片
15     try:
```

```

16         # 获取最近 24根 1h K线
17         ohlcv = okx.fetch_ohlcv(symbol, '1h', limit=24)
18         df = pd.DataFrame(ohlcv, columns=['ts', 'o', 'h', 'l', 'c',
19         'v'])
20
21         # 计算波动率 (最高-最低)/最低
22         df['vol'] = (df['h'] - df['l']) / df['l']
23         avg_vol = df['vol'].mean() * 100
24
25         results.append({'symbol': symbol, 'volatility': avg_vol})
26     except:
27         pass
28
29     # 排序并输出前 10 名
30     df_res = pd.DataFrame(results).sort_values(by='volatility',
31     ascending=False)
32     print("\n🔥 高波动币种推荐 (24h平均振幅):")
33     print(df_res.head(10))

```

2. 运行: `python scan_volatility.py`

3. 配置: 将排名靠前的币种填入 `config.json`。

方案 C: 调整策略参数 (AI 自主决策)

- 配置变更:
 - `timeframe`: 改为 5m (5分钟周期)。
 - `history_limit`: 改为 30 (减少历史负担, 专注近期)。
- AI 智能博弈 (Autonomous Mode):
 - **核心理念**: 彻底摒弃预设的刻板规则 (如 "RSI>70必卖"), 转而采用**全自主数据驱动**的决策模式。
 - **数据增强**: 投喂全量 OHLCV 数据 (含 Open/High/Low/Close), 使 AI 能识别 **Pin Bar (金针探底)**、**Engulfing (吞没形态)** 等关键 Price Action 信号。
 - **量价验证**: AI 会结合 **Volume (成交量)** 判断突破的有效性, 拒绝无量假突破。
 - **自我纠错**: AI 被赋予最高权限, 需时刻对比“当前持仓”与“最新盘面”。一旦发现方向错误 (如做空后遭遇大阳线逼空), AI 将**立即反手**, 承认错误并止损反向开仓。

方案 D: 适当增加杠杆

- 操作: 在 `config.json` 中将 `leverage` 从 2 提升到 5 或 10。
- 效果: 波动 1% 将带来 5%~10% 的盈亏变化。
- 警告: **盈亏同源**。高杠杆意味着爆仓风险剧增, 请务必配合更严格的止损 (`max_loss_rate` 设为 0.02 或更低)。

Q2: 为什么有时候机器人不交易?

- **震荡市 (Choppy Market)**: AI 判断 `ADX < 25`, 认为没有趋势, 为了防止磨损手续费, 建议 `HOLD` (观望)。这是正常的防守行为。
- **信心不足**: 信号信心为 `LOW`, 但配置要求 `MEDIUM`。
- **滑点拦截**: 价格波动太快, 超过了配置的 `max_slippage_percent`。

Q3: 为什么日志里全是 ZEC 的信息，看不到其他币？

这是日志打印机制导致的。

- **HOLD** 状态的币种很安静，只在最后的 **MARKET SCAN** 表格里显示一行。
- 正在尝试交易或报错的币种会打印大量调试信息，导致刷屏。
- **解决**: 只要在最后的表格里能看到币种，说明它就在运行。

Q4: 为什么我看 1分钟 (1m) 周期觉得不准？

- **噪音问题**: 1m 周期充满了随机漫步的噪音，假突破极多。
- **手续费磨损**: 1m 级别的微小利润往往覆盖不了双向手续费。
- **建议**: 坚持使用 **15m** 周期看趋势，配合 **15s** 轮询抓时机。

Q5: 怎么配置 API Key 最安全？

- **强烈建议**: 使用系统环境变量 (`OKX_API_KEY`, `DEEPSEEK_API_KEY`)。
- **当前机制**: 代码已强制开启安全模式。如果环境变量缺失，程序会拒绝启动。严禁将 Key 明文写在 `config.json` 里。

4. 版本历史备注 (Version Notes)

- **v3.2.0 (AI Evolution)**:
 - **架构升级**: 彻底重构了 AI 决策层 (`ai_strategy.py`)，移除所有硬编码规则，转型为 **全自主数据驱动 (Autonomous Data-Driven)** 模式。
 - **数据增强**: 向 AI 投喂全量 OHLCV 数据 (Open/High/Low/Close)，支持 Pin Bar、吞没等 Price Action 形态识别。
 - **纠错机制**: 赋予 AI 最高权限，可实时对比持仓与盘面，主动执行反向开仓以纠错。
 - **文档同步**: 更新了 `PROJECT_MANUAL.md` 和 `PROJECT_COMPLETE_FLOW.md` 以反映最新的 AI 决策逻辑。
- **v3.1.18 (Doc Update)**:
 - 新增了关于选币策略的 FAQ。
 - 优化了文档结构，增加了架构图。
- **v3.1.16**:
 - 修复了做空止损失效的致命 Bug。
 - 移除了 Watchdog 和移动止盈 (Trailing Stop) 等复杂功能，回归纯粹稳健的策略。
 - 优化了 AI Prompt，支持稳定币识别和杠杆感知。

文档生成时间: 2025-12-28