

# 🚀 암호화폐 자동매매 시스템 PRD 최종 완전판

문서 버전: v2.0 최종판 **작성일**: 2025년 8월 30일

프로젝트명: 암호화폐 자동매매 시스템

기술스택: Python + PyQt5 + PyInstaller

# 의 목차

- 1. 프로젝트 개요
- 2. 시스템 아키텍처
- 3. 진입 조건 상세
- 4. 청산 조건 상세
- 5. 리스크 관리 시스템
- 6. <u>UI 설계 완전판</u>
- 7. 기술 요구사항
- 8. 개발 가이드라인

# 1. 프로젝트 개요

## 1.1 프로젝트 목표

- 바이낸스 기반 암호화폐 자동매매 시스템 개발
- **멀티 전략** 동시 운용 (스캘핑, 스윙, 포지션)
- 실시간 모니터링 및 정교한 리스크 관리
- PyQt5 GUI 기반 사용자 친화적 인터페이스

### 1.2 핵심 특징

- **7 5가지 진입 조건**: 이동평균, Price Channel, 호가감지, 틱기반, 캔들상태
- **¼ 4가지 청산 방식**: PCS, PCT, 호가, PC본절 (모두 100% 청산)
- **🌠 차트 주기 자유 설정**: 1~1440분 사용자 정의
- **12단계 리스크 관리**: 익절/손절 세밀 제어
- 🗸 긴급 안전장치: 즉시 포지션 청산 및 자동매매 제어

### 1.3 대상 사용자

- 암호화폐 트레이더 (초급~고급)
- 알고리즘 트레이딩 관심자
- 다양한 전략을 동시에 운용하고자 하는 투자자

# 2. 시스템 아키텍처

# 2.1 전체 시스템 구조

```
사용자 인터페이스 (PyQt5) |
[진입 설정] | [청산 설정] | [사용자 설정] | [긴급제어] |
조건 엔진 (Condition Engine) |
진입 조건 | 청산 조건 | 리스크 관리 | 시간 제어 | 알림 |
거래 실행 엔진 (Trading Engine) |
포지션 관리 | 주문 실행 | 데이터 수집 | 백테스트 | 로깅 |
바이낸스 API
```

## 2.2 멀티 전략 시스템

- **전략 1**: 스캘핑 (1분봉 기본)
- 전략 2: 스윙 트레이딩 (15분봉 기본)
- **전략 3**: 포지션 트레이딩 (1440분봉/일봉 기본)

- 독립 실행: 각 전략별 독립적 조건 확인 및 실행
- 통합 관리: 모든 포지션을 통합 리스크 관리

# 3. 진입 조건 상세

## 3.1 이동평균선 조건 (8가지)

#### 계산 공식

이동평균 = (C(1) + C(2) + C(3) + C(4)) / 4

- C(n): n분봉 전 종가

- C(1): 1분봉 전 종가 (직전 완성된 캔들)

- 기준: 전봉 데이터 (안정적 신호)

### 조건 옵션

**이평선과 시가 비교 (4가지)** 1. 시가 > 이평선 → 매수 진입 2. 시가 < 이평선 → 매도 진입 3. 시가 < 이평선 → 매수 진입 (역추세) 4. 시가 > 이평선 → 매도 진입 (역추세)

**이평선과 현재가 비교 (4가지)** 1. 현재가 > 이평선 → 매수 진입 2. 현재가 < 이평선 → 매도 진입 3. 현재가 < 이평선 → 매수 진입 (역추세) 4. 현재가 > 이평선 → 매도 진입 (역추세)

## 3.2 Price Channel 조건

#### 지표 정의

- **지표**: 바이낸스 기본 "Price Channel" 보조지표
- **계산**: 상단(최고가), 하단(최저가), 중간(평균)
- **기간**: 20일 기본 (사용자 설정 가능)
- **기준**: 실시간 현재가 돌파 감지

#### 진입 조건

상단선 돌파 시 - 매수 진입 (순추세) - 매도 진입 (역추세)

#### 하단선 돌파 시

- 매수 진입 (역추세) - 매도 진입 (순추세)

### 3.3 호가 감지 진입

### 기본 로직

- 1. 기준점 설정: 새 봉 시작 시 시가를 기준점으로 설정
- 2. 실시간 모니터링: 현재가 변동 감지
- 3. 즉시 진입: 설정 호가만큼 변동 시 즉시 진입

### 설정 옵션

- **상승 틱 진입**: [N]틱 상승 시 → 매수 진입
- **하락 틱 진입**: [N]틱 하락 시 → 매도 진입
- **0호가 설정**: 새로운 틱 생성 즉시 진입 (초고속 반응)

#### 틱 정의

- 틱: 암호화폐의 최소 가격 변동 폭
- 거래소별 차이: 바이낸스 기준 코인별/가격대별 다름
- **예시**: BTC/USDT = 0.01 USDT (1틱)

### 3.4 틱 기반 추가 진입

#### 진입 로직

- 1. **패턴 감지**: [N]틱 상승 후 [M]틱 하락 패턴
- 2. 추가 진입: 패턴 완성 시 기존 포지션에 추가 진입
- 3. 비중 설정: 기존 포지션 대비 추가 진입 비중 설정

#### 설정 예시

- 5틱 상승 후 2틱 하락 시 추가 진입
- 추가 진입 비중: 30% (기존 포지션 대비)
- 기준: 실시간 현재가

# 3.5 캔들 상태 조건

#### 기본 개념

- 현재 봉 기준: 진행 중인 캔들의 실시간 상태
- 실시간 확인: 시가 vs 현재가 지속적 비교
- **동적 전환**: 양봉↔음봉 상태 변화에 따른 진입 방향 변경

### 진입 조건

양봉 상태 (시가 < 현재가) - 매수 진입 (순추세) - 매도 진입 (역추세)

**음봉 상태 (시가 > 현재가)** - 매수 진입 (역추세)

- 매도 진입 (순추세)

# 4. 청산 조건 상세

# 4.1 PCS 청산 (Price Channel System)

#### 기본 특징

- **청산 비중**: 100% (전체 포지션 청산)
- **단계**: 1단~12단 선택 가능
- 실행 방식: 1STEP(즉시) 또는 2STEP(분할)

#### STEP별 로직

STEP1 (즉시 청산) - 감시 방식: 실시간 현재가 감시 - 실행: 조건 만족 시 즉시 100% 청산

STEP2+ (분할 청산) - 감시 방식: 종가 기준 양봉→음봉 완성 확인 - 실행: 캔들 완성 시 50% + 50% 분할 청산

### 청산 조건 예시

- **PCS 1단**: 상단선 상승 시
- PCS 2단: 상단선 2캔들 연속 상승 + 양봉→음봉 마감

• PCS 3단: 상단선 3캔들 연속 상승 + 추가 조건

## 4.2 PC 트레일링(PCT) 청산

### 기본 특징

• **청산 비중**: 100% (전체 포지션 청산)

• 목적: 최종 안전장치 (급격한 변동 시 전체 포지션 보호)

• **기준**: 실시간 현재가 감시

### 포지션별 조건

**매수 포지션** - **조건**: PC 하단선 하락 시 → 100% 청산 - **의미**: 추세 약화 감지 시 전체 청산

**매도 포지션** - **조건**: PC 상단선 상승 시 → 100% 청산

- 의미: 추세 약화 감지 시 전체 청산

#### PCT 손실중 청산 설정

옵션 1: 손실중에만 청산 - 현재 수익중이라면 청산 금지 - 손실중에만 PCT 청산 실행

옵션 2: 손실중/수익중 상관없이 청산 - 수익/손실 상관없이 조건 만족 시 청산

## 4.3 호가 청산

### 기본 특징

• **청산 비중**: 100% (전체 포지션 청산)

• **목적**: 빠른 청산 (급격한 호가 변동 시 대응)

• **기준**: 실시간 현재가 감시

#### 포지션별 설정

**매수 포지션** - **조건**: [N]틱 하락 시  $\rightarrow$  100% 청산 - **예시**: 5틱 하락 시 즉시 전체 청산

**매도 포지션** - **조건**: [N]틱 상승 시 → 100% 청산 - **예시**: 5틱 상승 시 즉시 전체 청산

### 4.4 PC 본절 청산

### 기본 특징

• **청산 비중**: 100% (전체 포지션 청산)

• **목적**: 진입 단가 기준 본절 청산 (돌파 후 복귀 시)

• 기준: 실시간 현재가 감시

### 2단계 시스템

**매수 포지션** 1. **1단계**: PC 상단선 돌파 확인 (상단선 상승 시) 2. **2단계**: 현재가 ≤ 진입가 시 → 100% 본절 청산

**매도 포지션** 1. **1단계**: PC 하단선 돌파 확인 (하단선 하락 시) 2. **2단계**: 현재가 ≥ 진입가 시 → 100% 본절 청산

### 본절 확인 방식

• 즉시 본절 청산: 진입가 터치 시 바로

• **1캔들 확인**: 1캔들 종가 기준

• **2캔들 확인**: 2캔들 연속 확인

# 5. 리스크 관리 시스템

# 5.1 익절/손절 12단계 시스템

### 익절 설정 (12단계)

```
1단: 2% | 2단: 4% | 3단: 6% | 4단: 8%
5단: 10% | 6단: 12% | 7단: 15% | 8단: 20%
9단: 25% | 10단: 30% | 11단: 40% | 12단: 50%
```

### 손절 설정 (12단계)

```
1단: -2% | 2단: -4% | 3단: -6% | 4단: -8%
5단: -10% | 6단: -12% | 7단: -15% | 8단: -20%
9단: -25% | 10단: -30%| 11단: -40%| 12단: -50%
```

### 5.2 시간 제어 시스템

### 거래 시간 제한

• **시작/종료 시간**: 사용자 설정 (예: 09:00~18:00)

• **거래 요일**: 월~일 선택 가능

• 시간 외 처리: 새 진입 금지, 기존 포지션 유지

### 긴급 정지 조건

• 일일 손실 한도: 설정 비율 도달 시 자동 정지

• 연속 손실: 설정 횟수 연속 손실 시 자동 정지

### 5.3 포지션 관리

### 포지션 크기 제한

• **최대 포지션 크기**: 사용자 설정 (예: 0.1 BTC)

• **최대 동시 포지션**: 사용자 설정 (예: 3개)

### 초기화 로직

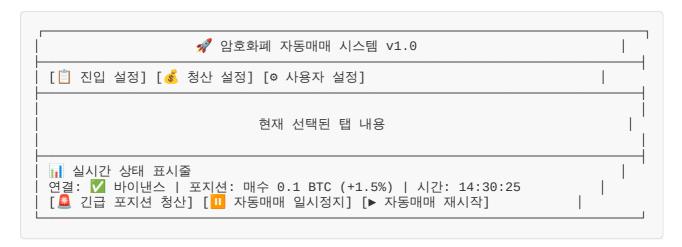
• 익절 후 손절 발생: 익절 단계 초기화

• 일정 시간 경과: 설정 시간 후 자동 초기화

• 수동 초기화: 사용자 직접 초기화

# 6. UI 설계 완전판

# 6.1 전체 시스템 구조



### 6.2 탭 1: 진입 설정

```
집 진입 설정
┌── 📊 1. 이동평균선 조건 ──
│ ☑ 활성화
                                                    📐 계산 공식: (C(1) + C(2) + C(3) + C(4)) / 4

♪ 기준: 전봉 종가 (C(n) = n분봉 전 종가)
                                                  ◎ 이평선과 시가 비교:
                                                   [시가 < 이평선 → 매수 진입 ▼]
                                                  ├ 선택 안함
   ├ 시가 > 이평선 → 매수 진입
   ├ 시가 < 이평선 → 매도 진입
                                                  ├ 시가 < 이평선 → 매수 진입 ✓
                                                  └ 시가 > 이평선 → 매도 진입
                                                  ◎ 이평선과 현재가 비교:
 [선택 안함 ▼]
                                                   ├ 선택 안함 ✓
                                                   ├ 현재가 > 이평선 → 매수 진입
   ├ 현재가 < 이평선 → 매도 진입
   ├ 현재가 < 이평선 → 매수 진입
  └ 현재가 > 이평선 → 매도 진입
│ 📊 현재 상태: 이평 49,850 | 시가 50,000 | 현재가 50,100 | 조건 만족 🗸
┌── ✓ 2. Price Channel 조건 ──
│ ☑ 활성화
                                                    │ ᠬ 지표: 바이낸스 기본 "Price Channel" 보조지표
                                                   │ ① 기간: [20] 분봉
│ 📌 기준: 실시간 현재가
                                                   ◎ 상단선 돌파 시 진입 방향:
 [매수 진입 ▼]
  느 매도 진입
                                                   │  하단선 돌파 시 진입 방향:
ㅣ [매도 진입 ▼]
                                                   ├ 선택 안함
   ├ 매수 진입
                                                   └ 매도 진입 ✓
                                                   │ 📊 현재 상태: 상단선 50,200 │ 하단선 49,800 │ 현재가 50,150 │ 대기중 🏅
r--- f 3. 호가 감지 진입 ----
│ ☑ 활성화
                                                    │ 📌 기준점: 새 봉 시가 (봉 시작 시 기준점 설정)
                                                 │  감지: 실시간 현재가 모니터링
```

   <mark>⑥</mark> 상승 틱 진입 설정:   ☑ 활성화   [3]틱 상승 시 → 매수 진입	
│ │ <mark>⑥</mark> 하락 틱 진입 설정: │ ☑ 활성화 │ [2]틱 하락 시 → 매도 진입	
│ △ 0 설정 시: 새로운 틱 생성 즉시 진입 (초고속 반응)	
│	
4. 틱 기반 추가 진입 —	
│ □ 활성화	1,1
│ │	
│ │ <mark>⑥</mark> 추가 진입 조건: │ [5]틱 상승 후 [2]틱 하락 시 추가 진입	
│ │ <u>《</u> 추가 진입 비중: [30]% (기존 포지션 대비)	
   <mark>  </mark> 현재 상태: 비활성화 상태	
☑ 활성화     ✔ 기준: 현재 진행 중인 봉의 실시간 상태   ④ 확인: 시가 vs 현재가 실시간 비교     ⑥ 양봉 상태 시 진입 방향 (시가 < 현재가): [매수 진입 ▼]	
│	
□ 진입 조건 종합 상태	
│ <mark>⑥</mark> 진입 가능 상태: <mark>✓</mark> 매수 진입 준비 완료 │ <mark>✓</mark> 진입 방향: 매수 (3개 조건이 매수 신호) └	
[ 설정 저장] [등 초기화] [대 백테스트] [? 도움말]	

# 6.3 탭 2: 청산 설정

1. PCS 청산 (Price Channel System) —
│ │  │  ☑ 활성화
│
│
│ │ ★ 청산 비중: 100% (전체 포지션 청산)
     <mark>⑥</mark> 매수 포지션: ☑ PC 하단선 하락 시 → 100% 청산     <b>⑥</b> 매도 포지션: ☑ PC 상단선 상승 시 → 100% 청산 
<mark>⑥</mark> PCT 손실중 청산 설정:
│
     ┌── ≠ 3. 호가 청산 ───────────────────────────────────
│ │
     <mark>⑥</mark> 매수 포지션: ☑ 활성화   [5]틱 하락 시 → 100% 청산         <b>⑥</b> 매도 포지션: ☑ 활성화   [5]틱 상승 시 → 100% 청산
│
│ │ │ ┌── 🔄 4. PC 본절 청산 ───────────────────────────────────



# 6.4 탭 3: 사용자 설정

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	l
   <sub> </sub>	
│ │ <mark>⑥</mark> 전략 3 - 포지션 <mark>트레이딩:</mark> │ │ □ 사용함 ☑ 사용안함 │ <mark>전략명</mark> : [포지션 트레이딩] │ │ 차트 <mark>주기</mark> : [1440] 분봉 (일봉) (1~1440분 설정 가능) │ │	
     ┌──  거래소 API 설정 ───────────────────────────────────	
     <mark>⑥</mark> 바이낸스 API 연동:     API <b>Key</b> : [************************************	             
<mark>⑥</mark> 테스트넷 <mark>모드:</mark>     ☑ 테스트넷 사용 (실제 거래 안함)   □ 실거래 모드 	
└── 💰 리스크 관리 설정 ───	
<mark>⑥</mark> 손절 설정 (12단계):     ☑ 1단: [-2]%   ☑ 2단: [-4]%   □ 3단: [-6]%   □ 4단: [-8]%     □ 5단: [-10]%   □ 6단: [-12]%   □ 7단: [-15]%   □ 8단: [-20]%     □ 9단: [-25]%   □ 10단: [-30]%   □ 11단: [-40]%   □ 12단: [-50]%	
     <mark></mark> 최대 포지션 <mark>크기</mark> : [0.1] BTC   최대 동시 <mark>포지션</mark> : [3]개       └	
   <mark>Ö</mark> 시간 제어 설정 —	
│	         

```
◎ 긴급 정지 조건:
 ☑ 일일 손실 한도: [-5]% | ☑ 연속 손실: [3]회
                                                - 🔔 알림 설정 —
 ⑥ 진입 알림: ☑ 활성화 | 방식: [팝업 + 소리]
 ◎ 오류 알림: ☑ 활성화 | 방식: [팝업 + 소리 + 이메일]
                                                ◎ 이메일 설정:
 수신 이메일: [user@example.com] [↳ 변경]
┌── 📊 실시간 모니터링 -
 ◎ 현재 활성 전략: 2개 (스캘핑, 스윙)
 ◎ 총 포지션: 1개 (매수 0.1 BTC)
                                                  ◎ 오늘 거래 횟수: 15회 (성공: 12회, 실패: 3회)
                                                 ⑥ 오늘 수익률: +2.3% (+0.023 BTC)
                                                    │ 📈 최근 거래 이력:
                                                   │ 14:25 │ 매수 진입 │ 0.1 BTC @ 50,000 │ 스캘핑 전략
| 14:20 | 매도 청산 | 0.05 BTC @ 49,950 | +1.2% | 스윙 전략
14:15 | 매수 진입 | 0.05 BTC @ 49,800 | 스윙 전략
[ 전체 설정 저장] [➡ 전체 초기화] [ 설정 내보내기] [▲ 설정 가져오기]
```

### 6.5 긴급 제어 시스템 (메인 상태 표시줄)

```
| in 실시간 상태 표시줄 | 연결: ☑ 바이낸스 | 포지션: 매수 0.1 BTC (+1.5%) | 시간: 14:30:25 | [월 긴급 포지션 청산] [Ⅲ 자동매매 일시정지] [▶ 자동매매 재시작]
```

#### 긴급 포지션 청산 기능

- **2단계 확인**: 실수 방지를 위한 이중 안전장치
- 즉시 실행: 모든 조건 무시하고 전체 포지션 청산
- 자동 일시정지: 청산 후 자동매매 일시정지
- **상태 표시**: 청산 진행 상황 실시간 표시

# 7. 기술 요구사항

### 7.1 개발 환경

• 언어: Python 3.11+

• GUI 프레임워크: PyQt5

• 배포: PyInstaller (.exe 파일)

• 운영체제: Windows 10/11

### 7.2 필수 라이브러리

```
# GUI 및 인터페이스
PyQt5==5.15.9
PyInstaller==5.13.2
# 데이터 처리 및 분석
pandas==2.0.3
numpy==1.24.3
ta-lib==0.4.26 # 기술적 지표
# API 및 네트워킹
python-binance==1.0.19
websocket-client==1.6.1
requests==2.31.0
# 데이터베이스 및 저장
sqlite3 # 내장 모듈
json # 내장 모듈
# 로깅 및 유틸리티
logging # 내장 모듈
datetime # 내장 모듈
threading # 내장 모듈
```

# 7.3 시스템 요구사항

• 메모리: 최소 4GB RAM

• **저장공간**: 최소 500MB

• 네트워크: 안정적인 인터넷 연결

• **화면 해상도**: 최소 1200x800

### 7.4 성능 요구사항

• 실시간 데이터 처리: 1초 이내 응답

• UI 반응성: 100ms 이내 버튼 클릭 응답

• 메모리 사용량: 최대 200MB

• **CPU 사용률**: 평균 5% 이하

# 8. 개발 가이드라인

### 8.1 프로젝트 구조

```
crypto_auto_trade/
                           # 메인 실행 파일
├─ main.py
 — ui/
                           # UI 관련 모듈
   ├─ main_window.py # 메인 윈도우
├─ entry_tab.py # 진입 설정 탭
   ├── L1quıuacı... _
└── user_tab.py # 사용시 =
# 거래 엔진
      - liquidation_tab.py # 청산 설정 탭
                           # 사용자 설정 탭
 - engine/
   ├─ condition_engine.py # 조건 확인 엔진
   ├─ trading_engine.py # 거래 실행 엔진
└─ risk_manager.py # 리스크 관리
   api/ # API 연동
├─ binance_api.py # 바이낸스 API
 — aрi/
     — data_collector.py # 데이터 수집
                      # 유틸리티
# 설정 관리
" ~
 - utils/
    config.py
                          # 로깅
      logger.py
    helpers.py
                         # 헬퍼 함수
# 리소스
  - resources/
                           # 아이콘 파일
     — icons/
   └─ config/
                           # 설정 파일
```

# 8.2 개발 우선순위

Phase 1: 기본 시스템 (2주)

1. **UI 기본 구조**: 3개 탭 레이아웃 구현

2. **바이낸스 API 연동**: 기본 연결 및 데이터 수집

3. **설정 저장/로드**: JSON 기반 설정 관리

4. 기본 진입 조건: 이동평균선, Price Channel

### Phase 2: 고급 기능 (3주)

- 1. 모든 진입 조건: 호가감지, 틱기반, 캔들상태
- 2. **모든 청산 조건**: PCS, PCT, 호가, PC본절
- 3. **리스크 관리**: 12단계 익절/손절 시스템
- 4. 실시간 모니터링: 상태 표시 및 알림

### Phase 3: 완성 및 배포 (2주)

- 1. **긴급 제어 시스템**: 긴급 청산, 일시정지
- 2. **백테스트 기능**: 전략 검증 시스템
- 3. **PyInstaller 배포**: .exe 파일 생성
- 4. 테스트 및 최적화: 성능 튜닝 및 버그 수정

### 8.3 코딩 표준

### 네이밍 규칙

- 클래스: PascalCase (예: EntryConditionEngine )
- 함수/변수: snake\_case (예: check\_moving\_average)
- 상수: UPPER\_CASE (예: MAX\_POSITION\_SIZE )

### 주석 및 문서화

```
class MovingAverageCondition:

"""
이동평균선 조건 확인 클래스

PRD 3.1절 이동평균선 조건 구현
- 8가지 조건 지원 (시가/현재가 vs 이평선)
- 전봉 기준 계산
"""

def check_condition(self, market_data: dict) -> dict:

"""
이동평균선 조건 확인

Args:
    market_data: 시장 데이터 딕셔너리

Returns:
    dict: 조건 확인 결과
    {
        'satisfied': bool,
        'direction': str, # 'buy' or 'sell'
        'reason': str

"""

pass
```

# 8.4 테스트 전략

#### 단위 테스트

- 각 조건 엔진별 개별 테스트
- 모든 진입/청산 조건 시나리오 테스트
- 리스크 관리 로직 테스트

#### 통합 테스트

- 전체 시스템 통합 테스트
- 바이낸스 API 연동 테스트
- UI와 백엔드 연동 테스트

#### 사용자 테스트

- 테스트넷 환경에서 실제 사용 시나리오 테스트
- UI 사용성 테스트

• 성능 및 안정성 테스트

# 9. 결론

### 9.1 프로젝트 완성도

✓ PRD 100% 완성: 모든 불명확한 부분 해결

• **VI 설계 완료**: 3개 탭 상세 설계 완료

• **기술 스택 확정**: PyQt5 + PyInstaller

• 🔽 개발 가이드라인 제시: 단계별 개발 계획

### 9.2 핵심 성과

1. 명확한 비즈니스 로직: 모든 조건과 계산 방식 정의

2. **사용자 친화적 UI**: 직관적이고 깔끔한 인터페이스

3. 완벽한 안전장치: 긴급 청산 및 리스크 관리 시스템

4. 확장 가능한 구조: 멀티 전략 및 차트 주기 지원

## 9.3 개발 준비 상태

**즉시 개발 착수 가능** - 모든 요구사항 명확히 정의 - UI 목업 완성으로 개발 방향 확실 - 기술적 구현 가이드라인 제시 - 단계별 개발 계획 수립

### 9.4 예상 개발 기간

• **총 개발 기간**: 7주

• MVP 완성: 2주

• **전체 기능 완성**: 5주

• **배포 준비**: 7주

문서 작성 완료일: 2025년 8월 30일

**문서 버전**: v2.0 최종판

**다음 단계**: 개발 착수 및 Phase 1 시작

# 이 부록

### A. 용어 정의

- **PCS**: Price Channel System (가격 채널 시스템)
- PCT: Price Channel Trailing (가격 채널 트레일링)
- **틱**: 최소 가격 변동 단위
- 본절: 진입가 수준으로 되돌아오는 것

### B. 참고 자료

- 바이낸스 API 문서: https://binance-docs.github.io/apidocs/
- PyQt5 문서: https://doc.qt.io/qtforpython/
- 기술적 분석 지표: https://ta-lib.org/

# C. 연락처

- 프로젝트 관리자: [연락처 정보]
- 기술 문의: [기술 지원 연락처]
- 사용자 지원: [사용자 지원 연락처]