

MT4CTP 简版终端使用文档

一、简要说明

- MT4CTP 软件终端是基于 QT5.14.2 版本开发的，由于 ubuntu 发行版对多屏幕驱动支持不是太好，故只提供 windows 版本。
- MT4CTP 终端最低运行环境：
 - CPU：至少双核 CPU，主频在 2Ghz 以上
 - 内存：4GB 或以上
 - 硬盘：视使用情况而定，落地期货期权快照行情需要较大的硬盘，机械硬盘或固态硬盘均可。不进行快照行情落地只需要少量空间即可。
- 操作系统:win7/win8/win10/win11 64bit 。不支持 32 位操作系统。
- 终端支持功能:

- 自定义指标、策略、预警，手动报撤单、柜台条件单、柜台预埋单。多屏显示、多屏联动。所有期货合约和大部分期权合约。
- 注意:该终端只支持中低频交易，不支持高频交易。个人认为从下单到成交响应完成的时间间隔不是微秒级别都不是高频交易，高频交易与低延迟环境还需要区分一下的。

二、用户名和密码

- 终端使用以前需要进行注册，注册成功以后终端才可使用。为了防止频繁注册，注册时终端会采集 MAC 地址。注册、账号验证、改密通过 passwdmanger.exe 实现。注册完成账号以后把 config724.txt 和 confighy.txt 文件中的账号密码改为自己的 simnow 账号密码即可。
- MT4CTP.exe 支持实时交易实时 K 线生成，MT4CTP724.exe 只支持 7*24 小时环境。

三、策略部分

- 策略部分的编写是基于 qt 插件的。策略编写主要基于三个头文件接口，分别是 strategyinterface.h、strategyinterfaceopt.h 以及 strategyinterfacemix.h。下面分别对各个头文件进行说明。有关 qt 插件编写的知识请问度娘。期货单合约策略的含义是接口函数的某些参数只针对当下接收到行情的合约，而不是此接口只能用于某个固定的合约，相反，可以配置多个合约都以这个接口运行。期权单合约策略也是如此。多合约策略接口的函数参数由于是全市场的，故可以配置套利策略，理论上策略可以有 N 个腿。

(1) 期货单合约策略

strategyinterface.h 文件中虚函数参数说明如下图

```
1  #ifndef SINGLEIDSTRAINTERFACEFUT_H
2  #define SINGLEIDSTRAINTERFACEFUT_H
3  返回策略的名字，终端内主要用此名字区分不同的策略，名字在终端内唯一
4  #include <QObject>
5  #include "pluginshareddatastruct.h"
6
7  #define StrategyInterface_IID "plugin.strategyinterface"
8
9
10
11 class StrategyInterface
12 {
13 public:
14     virtual ~StrategyInterface(){}
15     virtual QString getFileName() = 0;
16     virtual strategyRet funStrategyCal(QHash<QString,QHash<QString,QHash<int,QVector<double>>>> &indicatordat,KLine *p,QVariant &medbuf,QVector<int> &barscountAtPrevTickList,
17                                     TThostFtdcMoneyType Balance,TThostFtdcMoneyType CurrMargin,TThostFtdcMoneyType Available,TThostFtdcMoneyType AllFrozenMargin,
18                                     QList<pendingorder> &pendList,QList<posttotalfieldfuture> &longposList,QList<posttotalfieldfuture> &shortposList,
19                                     QList<traderecord> &lastBuyOpenRecList,QList<traderecord> &lastSellOpenRecList,
20                                     QString straid,TThostFtdcVolumeMultipleType VolumeMultiple,TThostFtdcPriceType PriceTick,TThostFtdcRatioType OpenRatioByMoney,
21                                     TThostFtdcMoneyType OpenRatioByVolume,TThostFtdcRatioType CloseRatioByMoney,TThostFtdcMoneyType CloseRatioByVolume,
22                                     TThostFtdcRatioType CloseTodayRatioByMoney,TThostFtdcRatioType CloseTodayRatioByVolume,
23                                     CThostFtdcDepthMarketDataField &tickdata) = 0;
24
25 };
26
27 QT_BEGIN_NAMESPACE
28 Q_DECLARE_INTERFACE(StrategyInterface, StrategyInterface_IID)
29 QT_END_NAMESPACE
30
31 #endif // SINGLEIDSTRAINTERFACEFUT_H
```

对应指标中的各项的名字比如macd中的dea,diff,histogram

对应指标的名字比如macd

时间周期:0对应1分钟, 1对应3分钟, 2对应5分钟, 3对应8分钟, 4对应10分钟, 5对应15分钟, 6对应30分钟, 7对应60, 8对应120, 9对应240, 10对应日线

该时间周期下的指标向量

K线数据结构指针-p[0]对应1分钟K线数据, p[1]对应3分钟K线数据, 依次类推.....

策略运行过程中各种中间缓冲值的保存

上一个tick到来后各时间周期下的K线数

Balance->当前账户的动态权益

CurrMargin->当前账户的总保证金占用

Available->当前账户的总可用资金

AllFrozenMargin->当前账户的总冻结资金

pendList->当下合约报单未成交列表

longposList->当下合约的多头持仓汇总表

shortposList->当下合约的空头持仓汇总表

lastBuyOpenRecList->当下合约的多开持仓明细列表

lastSellOpenRecList->当下合约的空开持仓明细列表

straid->该策略对应的id, 此id可用于区分持仓明细列表中哪些是此策略发报的单子, 也可区分未成交列表中的报单

VolumeMultiple->合约乘数

PriceTick->价格变动刻度

OpenRatioByMoney一直到CloseTodayRatioByVolume->请参考CTPAPI

tickdata->最新的快照数据

(2) 期权单合约策略

strategyinterfaceopt.h 是期权单合约策略的接口头文件，大部分都和期货单合约策略的参数相同，不同的地方使用下图标注

```
1  #ifndef SINGLEIDSTRAINTERFACEOPT_H
2  #define SINGLEIDSTRAINTERFACEOPT_H
3
4  #include <QObject>
5  #include "pluginshareddatastruct.h"
6
7  #define StrategyInterfaceOpt_IID "plugin.strategyinterfaceopt"
8
9
10
11 class StrategyInterfaceOpt
12 {
13 public:
14     virtual ~StrategyInterfaceOpt(){}
15     virtual QString getFileName() = 0;
16     virtual strategyRet funStrategyCal(QHash<QString,QHash<QString,QHash<int,QVector<double>>>> &indicatordat,KLineOption *p,QVariant &medbuf,QVector<int> &barscountAtPrevTickList,
17                                     TThostFtdcMoneyType Balance,TThostFtdcMoneyType CurrMargin,TThostFtdcMoneyType Available,TThostFtdcMoneyType AllFrozenMargin,
18                                     QList<pendingorder> &pendList,QList<posttotalfieldoption> &longposList,QList<posttotalfieldoption> &shortposList,
19                                     QList<traderecord> &lastBuyOpenRecList,QList<traderecord> &lastSellOpenRecList,
20                                     QString straid,TThostFtdcVolumeMultipleType VolumeMultiple,TThostFtdcPriceType PriceTick,TThostFtdcRatioType OpenRatioByMoney,
21                                     TThostFtdcMoneyType OpenRatioByVolume,TThostFtdcRatioType CloseRatioByMoney,TThostFtdcMoneyType CloseRatioByVolume,
22                                     TThostFtdcRatioType CloseTodayRatioByMoney,TThostFtdcRatioType CloseTodayRatioByVolume,
23                                     TThostFtdcRatioType StrikeRatioByMoney,TThostFtdcRatioType StrikeRatioByVolume,
24                                     CThostFtdcDepthMarketDataField &tickdata) = 0;
25
26 };
27
28 QT_BEGIN_NAMESPACE
29 Q_DECLARE_INTERFACE(StrategyInterfaceOpt, StrategyInterfaceOpt_IID)
30 QT_END_NAMESPACE
31
32 #endif // SINGLEIDSTRAINTERFACEOPT_H
```

该账户下的权利金收支

此处与期货单合约策略接口不同，为了减少降维打击，期权没有合成太小的周期，也为了节省终端所占用的内存
0->3min 1->8min 2->15min 3->30min 4->60min 5->120min 6->240min 7->日线

该账户下的权利金冻结金额

请注意此处与期货单合约的KLine结构有所不同，少了快照流的统计数据

请参考CTPAPI

(3) 多合约策略

strategyinterfacemix.h 是多合约策略的接口头文件，各参数说明如下图

```
1 #ifndef MULTIIDSTRAINTERFACE_H
2 #define MULTIIDSTRAINTERFACE_H
3
4 #include <QObject>
5 #include "pluginsharedatastruct.h"
6
7 #define StrategyInterfaceMix_IID "plugin.strategyinterfacemix"
8 返回策略名字。此策略名字在该终端下唯一
9
10
11 class StrategyInterfaceMix
12 {
13 public:
14     virtual ~StrategyInterfaceMix(){}
15     virtual QString getFileName() = 0;
16     //所有参数都是针对全市场
17     virtual strategyRet funStrategyCal(QHash<QString,struct InstrumentOrder > &futuredata,QHash<QString,struct optionInstrInfo > &optiondata,QVariant &medbuf,
18         TThostFtdcMoneyType Balance,TThostFtdcMoneyType CurrMargin,TThostFtdcMoneyType Available,TThostFtdcMoneyType AllFrozenMargin,
19         TThostFtdcMoneyType AllCashIn,TThostFtdcMoneyType AllFrozenCashIn,pendingorder *pPendingOrder,int PendingOrderSize,
20         posttotalfieldfuture *pPosTotalFutField,int futpostotalsize,postotalfieldoption *pPosTotalOptField,int optpostotalsize,
21         traderecord *pTradeRecord,int TradeRecordSize,traderecord *pHistoryTradeRecord,int HistoryTradeRecordSize,
22         QString straid,CTHostFtdcDepthMarketDataField &tickdata) = 0;
23 };
24
25 QT_BEGIN_NAMESPACE
26 Q_DECLARE_INTERFACE(StrategyInterfaceMix, StrategyInterfaceMix_IID)
27 QT_END_NAMESPACE
28
29 #endif // MULTIIDSTRAINTERFACE_H
30
```

期货合约的代码比如i2309/rb2310/IF2309

对应合约下的数据，包含合约基础参数数据和行情数据

期权合约的代码

对应该期权合约下的基础合约参数以及行情数据

该策略运行时的中间数据buff

straid->该策略对应的策略id,方便查找该策略的操作记录

tickdata->该策略目前在合约的最新快照数据

pPendingOrder->指向当前账户下当前交易日所有报单列表，不限于未成交

PendingOrderSize->报单列表的总条数

pPosTotalFutField->指向当前账户下的期货持仓数据汇总，可能有已平仓的或挂单未成交的记录

futpostotalSize->期货持仓数据汇总的条数

pPosTotalOptField->指向当前账户下的期权持仓数据汇总，可能有已平仓的或挂单未成交的记录

optpostotalSize->期权持仓数据汇总的条数

pTradeRecord->指向当前交易日该账号下的所有成交记录，包含期货期权

TradeRecordSize->所有成交记录的条数

pHistoryTradeRecord->上个交易日结算后持仓的成交记录

HistoryTradeRecordSize->昨持仓成交记录的条数

(4) 编程注意事项

对于(1)(2)(3)接口中的 medbuf 参数，建议编写时采用定长机制，策略自行维护长度并更新中间缓冲数据。特别是对于多合约策略更应该如此。

对于(1)(2)接口中的 indicatordat 参数，此指标参数只考虑定制期权和定制期货范围内的合约，在范围外的合约此参数为空

对于和 K 线长度等长的中间缓冲数据向量，建议做成指标插件，除非该合约不在定制期权和定制期货范围内

四、指标部分

指标部分的编写也是基于 QT 插件的，终端默认只对定制期权和定制期货范围内的合约进行指标图形显示。如需对范围外的合约进行自定义指标计算，请在策略接口中进行计算并将结果缓存在 medbuf 参数中。

indicatorinterface.h 接口头文件用于期货自定义指标，indicatorinterfaceoption.h 接口头文件用于期权自定义指标。

由于此部分设计界面显示，接口较多且繁杂，故归类如下图

```
class IndicatorInterface
{
public:
    virtual ~IndicatorInterface(){}
    virtual QString getFileName() = 0;
    virtual indicatorParam initIndicatorBuffer() = 0; //获取指标的名称buffer类别以及输入参数
    virtual void calIndicatorData(KLine &kl,int period,QVector<double> &inputgram,QHash<QString,QHash<int,QVector<double>>> &indicatorbuffer,bool isallcal) = 0; //计算指标各周期数据 //a164
    virtual struct aboutIndiPlot initUiPlot(QCustomPlot *customplot,QHash<QString,QHash<int,QVector<double>>> &indicatorbuffer,QVector<double> &inputpram,
        int axisRectIndex,QCPMarginGroup *group) = 0; //初始化指标的显示风格并返回各个plot的指针
    virtual void updatePlot(QHash<QString,QHash<int,QVector<double>>> &indicatorbuffer,QHash<QString,QVector<QCPAbstractPlottable * > > &plotOfIndicator,QVector<double> &inputgram,
        int period,bool isallupdated) = 0; //更新plot
};
```

通用接口

指标计算接口

指标显示接口

这里只以期货自定义指标接口示例，期权自定义指标接口是同样的分类。如果不需要界面显示自定义指标，只需关注指标计算接口就可以了。由于期货期权自定义指标接口头文件相似度很高，这里只以期货头文件接口进行参数说明。

```
7 #define IndicatorInterface_IID "plugin.indicatorinterface"
8
9 struct indicatorParam
10 {
11     QPair<QStringList,QVector<double>> nameOrParam;
12 };
13 返回指标插件的名字，该名字在期货自定义指标名字集合中唯一
14
15 struct aboutIndiPlot
16 {
17     QPair<QString,QHash<QString,QVector<QCPAbstractPlottable * > > > indicatorplots;
18     QPair<QCPAxisRect *,bool> updateAxisrect;
19 };
20
21 class IndicatorInterface
22 {
23 public:
24     virtual ~IndicatorInterface(){}
25     virtual QString getFileName() = 0;
26     virtual indicatorParam initIndicatorBuffer() = 0; //获取指标的名称buffer类别以及输入参数
27     virtual void calIndicatorData(KLine &kl,int period,QVector<double> &inputgram,QHash<QString,QHash<int,QVector<double>>> &indicatorbuffer,bool isallcal) = 0; //计算指标各周期数据 //a164
28     virtual struct aboutIndiPlot initUiPlot(QCustomPlot *customplot,QHash<QString,QHash<int,QVector<double>>> &indicatorbuffer,QVector<double> &inputpram,
29         int axisRectIndex,QCPMarginGroup *group) = 0; //初始化指标的显示风格并返回各个plot的指针
30     virtual void updatePlot(QHash<QString,QHash<int,QVector<double>>> &indicatorbuffer,QHash<QString,QVector<QCPAbstractPlottable * > > &plotOfIndicator,QVector<double> &inputgram,
31         int period,bool isallupdated) = 0; //更新plot
32 };
33
34 QT_BEGIN_NAMESPACE
35 Q_DECLARE_INTERFACE(IndicatorInterface, IndicatorInterface_IID)
36 QT_END_NAMESPACE
37
38 #endif // INDICATORINTERFACE_H
39
```

指标名字以及参数的初始化

当下合约某个时间周期的k线结构数据

时间周期比如0对应1min周期，1对应3min周期，.....

该指标的输入参数比如macd指标的12,26,9

策略接口头文件中的indicatordat参数来源于此

false->只取最后一个作为计算结果

true->计算K线整个数据向量

指标显示接口各个参数的含义请参见 ma. dll 和 macd. dll 的源码，解释起来比较复杂，ma. dll 是简单移动平均均线的实现，绘在主图上，macd. dll 是 MACD 的实现，绘在副图上。只要对 QCustomPlot 类有一定了解，源码理解起来不难。

indicatorinterfaceoption.h 期权自定义指标接口头文件与期货的类似。

指标和策略的计算都可引入任意的第三方库，把第三方库的 dll 放在 MT4CTP 文件夹下，就可保证自定义指标或策略的正常运行。以下是两个指标库地址：

<https://tulipindicators.org/>

<https://www.ta-lib.org/>

本人喜欢使用第一个，运行效率高且很容易理解。

五、预警部分

预警部分比较简单，只有两个参数，indicatordat 参数来源于自定义指标接口，medbuff 是预警计算过程中的中间 buff 缓冲。分为期货预警和期权预警。期货预警对应着 AlertInterface.h 接口头文件，期权预警对应着 AlertInterfaceOpt.h 头文件。可在下图界面打开声音提醒功能



另外，alertRet 结构中的 context 内容不要超过三项，否则会不显示三项以外的内容，此三项一般为:多空方向、预警的时间周期、自定义的预警内容。

六、插件安装目录

插件编译好以后需要放在特定目录才能正常运行，如下图



七、终端目录说明

下面的图对重要目录或文件进行说明

此电脑 > 本地磁盘 (F:) > MT4CTP

名称	修改日期	类型	大小	
alert	2023/5/20 18:52	文件夹		盘中预警信息输出文件夹
flow	2023/5/20 18:47	文件夹		
flowqry	2023/5/20 18:47	文件夹		
iconengines	2023/5/20 18:47	文件夹		
imageformats	2023/5/20 18:47	文件夹		
kline	2023/5/20 18:48	文件夹		期货期权K线存储文件
optionkline	2023/5/20 18:51	文件夹		
platforms	2023/5/20 18:51	文件夹		
plugins	2023/5/20 18:51	文件夹		插件安装目录
sqldrivers	2023/5/20 18:51	文件夹		每个交易日成交记录保存
styles	2023/5/20 18:51	文件夹		
traderecord	2023/5/20 18:51	文件夹		自动化启动脚本
translations	2023/5/20 18:51	文件夹		
ayto.bat	2023/4/10 17:23	Windows 批处理...	1 KB	期货账号配置文件
confighy.txt	2023/5/20 18:57	文本文档	1 KB	期货定制合约保存文件--每个交易日更新
customlds.txt	2023/5/19 15:16	文本文档	3 KB	
customldsbak.txt	2023/1/16 1:24	文本文档	2 KB	
D3Dcompiler_47.dll	2014/3/11 18:54	应用程序扩展	4,077 KB	自动化停止脚本
freelut.dll	2015/3/14 16:02	应用程序扩展	230 KB	主力合约保存文件---每个交易日自动更新
kill.exe.bat	2023/4/10 17:33	Windows 批处理...	1 KB	
libEGL.dll	2020/3/27 21:18	应用程序扩展	24 KB	行情与快照落地配置文件
libGLSv2.dll	2020/3/27 21:18	应用程序扩展	3,491 KB	
mainlds.txt	2023/5/19 15:25	文本文档	1 KB	终端可执行文件
mdconfig.txt	2023/5/11 20:42	文本文档	1 KB	
MT4CTP.exe	2023/5/21 9:55	应用程序	1,253 KB	MT4CTP终端注册小工具
opengl32sw.dll	2016/6/14 20:00	应用程序扩展	20,433 KB	
passwdmanger.exe	2023/5/20 19:05	应用程序	63 KB	
qcustomplot2.dll	2022/12/26 14:15	应用程序扩展	1,044 KB	
qcustomplot2.lib	2022/12/26 14:15	Object File Library	917 KB	
Qt5Core.dll	2020/3/27 21:18	应用程序扩展	6,049 KB	
Qt5Gui.dll	2020/3/27 21:18	应用程序扩展	6,963 KB	
Qt5Network.dll	2020/3/27 21:18	应用程序扩展	1,350 KB	
Qt5OpenGL.dll	2020/3/27 21:18	应用程序扩展	321 KB	
Qt5OpenGLd.dll	2020/3/27 21:18	应用程序扩展	750 KB	
Qt5PrintSupport.dll	2020/3/27 21:18	应用程序扩展	319 KB	
Qt5Sql.dll	2020/3/27 21:18	应用程序扩展	208 KB	
Qt5Svg.dll	2020/3/28 2:41	应用程序扩展	330 KB	

MT4CTP.exe	2023/5/21 9:55	应用程序	1,253 KB	
opengl32sw.dll	2016/6/14 20:00	应用程序扩展	20,433 KB	行情窗口绘制库
passwdmanger.exe	2023/5/20 19:05	应用程序	63 KB	
qcustomplot2.dll	2022/12/26 14:15	应用程序扩展	1,044 KB	
qcustomplot2.lib	2022/12/26 14:15	Object File Library	917 KB	
Qt5Core.dll	2020/3/27 21:18	应用程序扩展	6,049 KB	
Qt5Gui.dll	2020/3/27 21:18	应用程序扩展	6,963 KB	
Qt5Network.dll	2020/3/27 21:18	应用程序扩展	1,350 KB	
Qt5OpenGL.dll	2020/3/27 21:18	应用程序扩展	321 KB	
Qt5OpenGLd.dll	2020/3/27 21:18	应用程序扩展	750 KB	
Qt5PrintSupport.dll	2020/3/27 21:18	应用程序扩展	319 KB	
Qt5Sql.dll	2020/3/27 21:18	应用程序扩展	208 KB	
Qt5Svg.dll	2020/3/28 2:41	应用程序扩展	330 KB	
Qt5Widgets.dll	2020/3/27 21:18	应用程序扩展	5,406 KB	
selflds.txt	2023/5/19 15:16	文本文档	1 KB	期货期权结算价保存文件
selfldsbak.txt	2023/5/11 15:16	文本文档	1 KB	
selfoptionlds.txt	2023/5/19 15:16	文本文档	0 KB	
settlementprice.txt	2023/5/19 15:40	文本文档	11 KB	策略配置文件
settlementpriceoption.txt	2023/5/19 15:40	文本文档	185 KB	
strategyscope.txt	2023/4/19 22:08	文本文档	0 KB	
thostmduserapi_se.dll	2021/4/6 9:38	应用程序扩展	2,833 KB	
thostmduserapi_se.lib	2021/4/6 9:38	Object File Library	4 KB	指标库
thosttraderapi_se.dll	2021/4/6 9:36	应用程序扩展	3,206 KB	
thosttraderapi_se.lib	2021/4/6 9:36	Object File Library	4 KB	vc运行库可执行文件
tulip.dll	2022/12/26 14:22	应用程序扩展	232 KB	
tulip.lib	2022/12/26 14:22	Object File Library	41 KB	
vc_redist.x64.exe	2022/3/27 23:59	应用程序	14,953 KB	

八、配置文件

mdconfig.txt

mdconfig.txt

1 KlineSize=1500

2 OptionMdEnable=1

3 FutureTickLocal=1

4 OptionTickLocal=0

5 FutureMarginQry=0

6 FutureComisionQry=0

7 FutureOrderQry=0

8

当前交易日结束后终端保存的各周期最大K线数目，以方便下个交易日加载

期权行情使能项，1->使能期权行情 0->关闭期权行情

期货快照行情数据是否存盘，保存的是全市场期货合约行情数据 1->存盘 0->不保存快照数据

期权快照行情数据是否存盘，保存的是定制期权合约的行情数据 1->存盘 0->不保存快照数据

定制期货合约的保证金率/费查询使能项。1-使能查询 0->关闭查询

所有期货合约的手续费率/费查询使能项。1-使能查询 0->关闭查询

中金所合约申报费查询项。1-使能查询 0->关闭查询

如果硬盘空间较小，建议把存盘使能全部关闭

策略配置文件 strategyscope.txt。比如 esintermarket.dll 在螺纹、热卷期货上表现比较好，fedswing.dll 在苹果、棕榈期货上表现比较好，meanswing.dll 在锌期权上还可以，ruggierobond.dll 在橡胶期权、铜期权上表现优良。sledgehammer.dll 在纯碱和玻璃上套利表现好，我们可以在 strategyscope.txt 进行如下配置：

strategyscope.txt

1 EsinterMarket=01,rb2310\$hc2310

2 FedSwing=02,AP310\$p2309

3 meanswing=03,zn2306P20000

4 ruggierobond=04,ru2309C130000\$cu2308P63000

5 SledgeHammer=05,SA401\$FG309

EsinterMarket,FedSwing,meanswing,ruggierobond,SledgeHammer是策略接口函数getFileName的返回值，此处不一定非要和dll文件同名

01~05是策略接口中的straid参数，范围从01~99

插件的存放目录如下：

此电脑 > 本地磁盘 (F:) > MT4CTP > plugins > strategy

名称

esintermarket.dll

fedswing.dll

此电脑 > 本地磁盘 (F:) > MT4CTP > plugins > strategyopt

名称

meanswing.dll

ruggierobond.dll

此电脑 > 本地磁盘 (F:) > MT4CTP > plugins > strategymix

名称

sledgehammer.dll

九、辅助界面显示

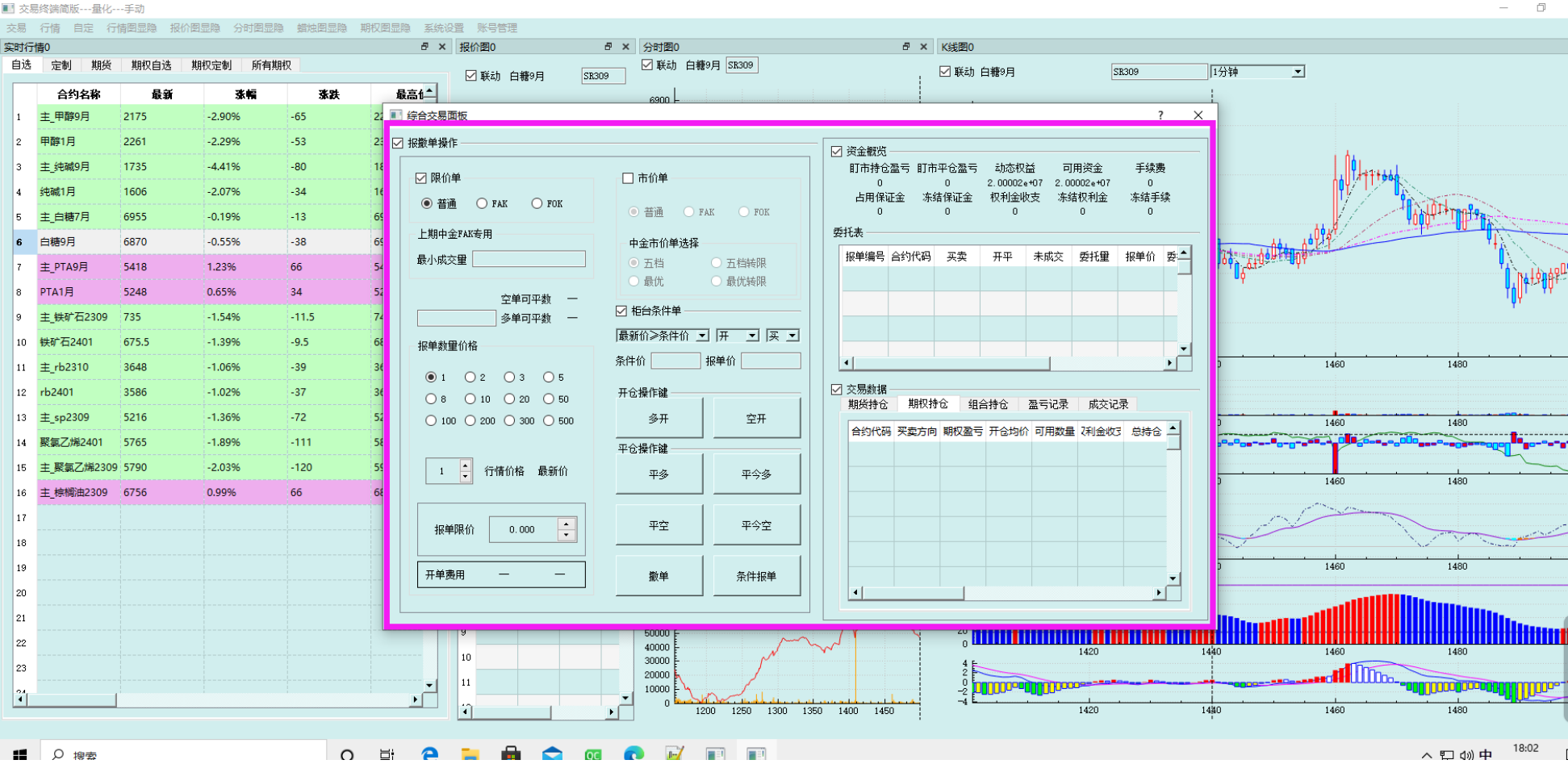
多屏界面



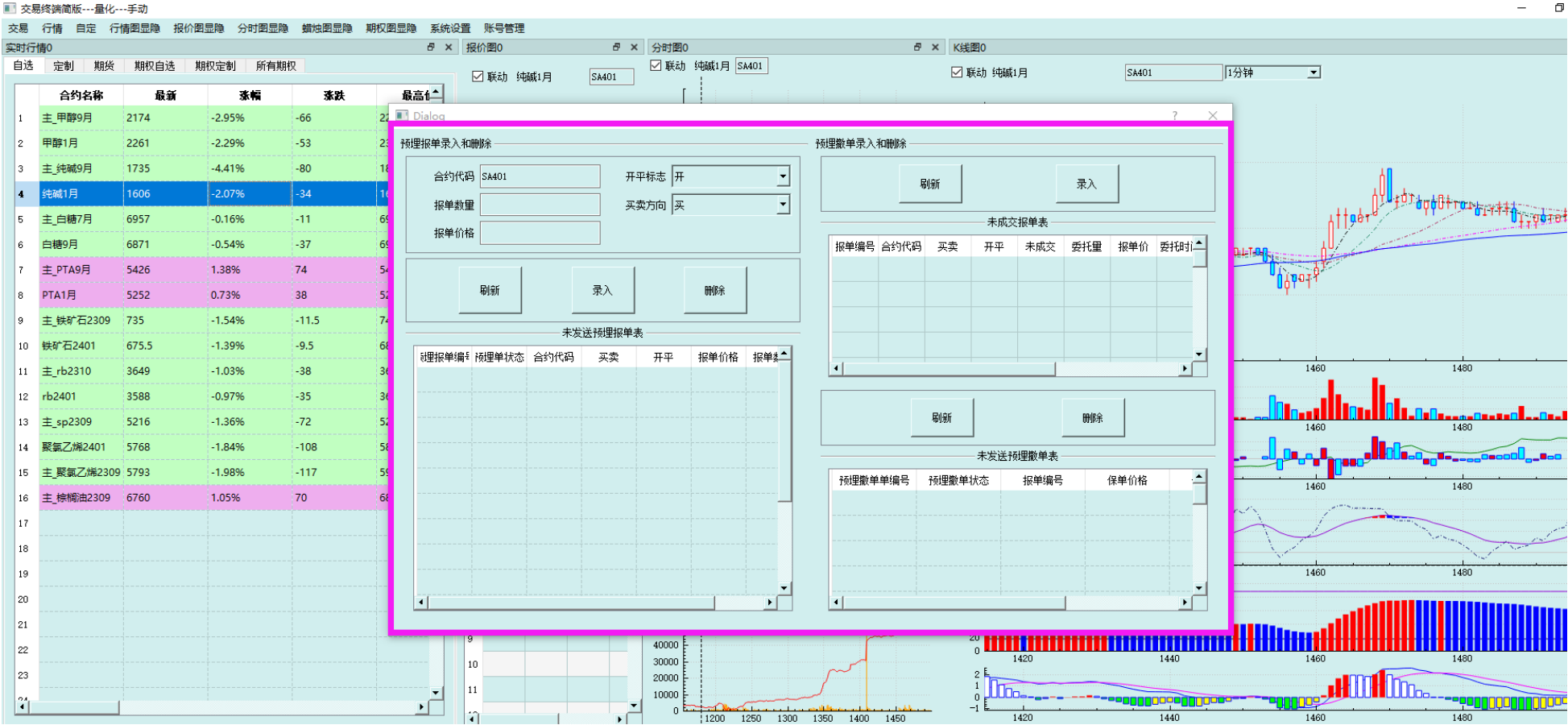
单屏界面



综合交易面板



预埋单面板



十、示例程序

策略示例参见\MT4CTP\example\strategy

指标示例参见\MT4CTP\example\indicator

预警示例参见\MT4CTP\example\alert

十一、回测终端

目前单合约单周期回测终端、单合约多周期终端已测试完毕，只支持快照级别回测，待文档整理出来会公布出来以供有缘人使用，多合约多周期回测终端目前正在开发中，时间同步问题是个不算小的问题。

欢迎大家一块学习交流，本人 QQ441855316