

○ 「 資産カーブをチャート上に表示する 」

- ・アメンボは、以前「擬似トレード提案（その1）」で、擬似トレード方法とチャート上への、「資産カーブ」表示例を投稿しました。
すると、当然ですが、今度は「実トレード」と「ストラテジー・テスター」のチャート上に「資産カーブ」を表示してみたくになりました。
- ・資料調査と試行錯誤の結果、とても簡単な方法に行き着きましたので報告いたします。
本稿では、「資産カーブ」を表示しましたが、「損益」などの他のデータを表示することも可能です。（当然ですね）

※本稿では、『チャート上への資産カーブ表示』に焦点を絞っていますので、
サンプルとして使用するEAは、たまたま検討を始めたばかりの物を使用しました、
ご了解ください。

（現状では資産カーブは右肩下がり続けるEAサンプルです）

しかしながら、一応「確認用サンプル」としてダウンロード用に準備しました。

※USDJPY 1M チャート上で実施

※「実トレード」上への「資産カーブ」描画は、「デモ・トレード」画面上で代用しています。

※ダウンロード用資料；

① 「myTest_SMA_03.mq4」

② 「Disp_Balance.mq4」

目次： <資産カーブ表示方法>

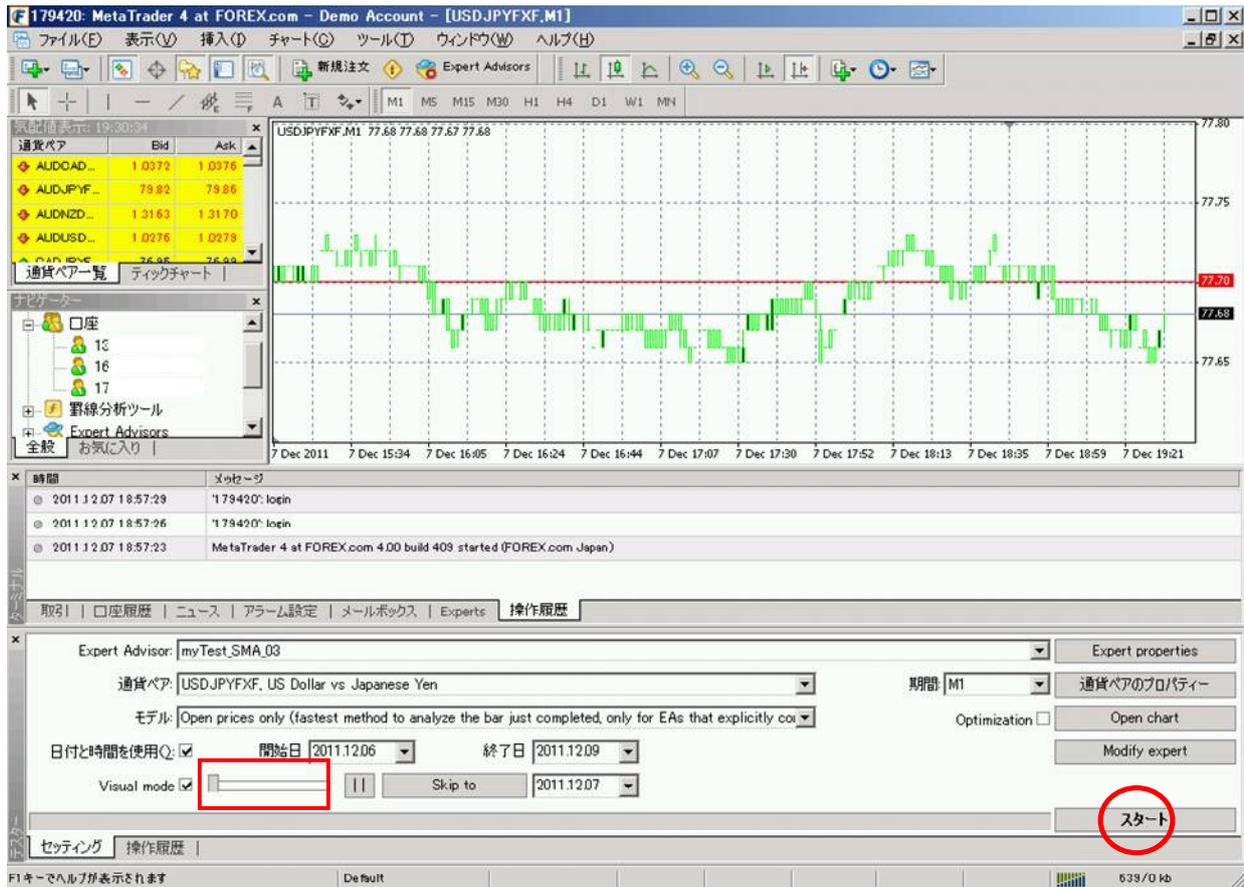
1. ストラテジー・テスター（visual mode）上での表示 . . . 2頁
 - （1）EA「myTest_SMA_03.mq4」をStrategy Testerにセットする
 - （2）ビジュアル・モードを開始・即ストップしてチャートを開く
 - （3）資産表示インディケータ「Disp_Balance.mq4」を、
ビジュアル・モード（Visual mode）のチャート上にセットする
 - （4）ビジュアル・モードを進めていく
2. リアルチャート（デモ・モード）上での表示 . . . 5頁
 - （1）資産表示インディケータ「Disp_Balance.mq4」を、
チャート上にセットする
 - （2）EA「myTest_SMA_03.mq4」をStrategy Testerにセットする
3. MQL4コード内容一覧 . . . 7頁
 - （1）動作確認用ストラテジー<EA> 「myTest_SMA_03.mq4」
 - （2）資産カーブ表示用<インディケータ> 「Disp_Balance.mq4」

<資産カーブ表示方法>

1. ストラテジー・テスター (visual mode) 上での表示

(1) EA「myTest_SMA_03.mq4」を Strategy Tester にセットする
まずは、通常と同じバックテストの準備をしておきます。

- 通貨ペア； USD JPY
- 期間； M1
- 日付と時間を使用 にチェック
- Visual Mode にチェック



(2) ビジュアル・モードを開始・即ストップしてチャートを開く

- Visual mode の「スライダー」を左の端に置いて、**スタート** を選択
⇒ Visual mode のチャートが表示されます。



※チャートを観易くするために、「背景を白色、枠線を黒色、ローソク足」に変更する



(3) 資産表示インディケータ「Disp_Balance.mq4」を、
ビジュアル・モード (Visual mode) のチャート上にセットする



(4) ビジュアル・モードを進めていく

・ Visual Mode のスライダーを少し右の方へ動かす⇒ テスト開始、「資産カーブ」も描画



※テストが更に進行していきます



※ [日付と時間を使用] の [終了日] を例えば明日にしておくと、現時点まで達したところでストップ、テスト完了となります。

The screenshot shows the MetaTrader 4 interface. The main window displays a candlestick chart for USDJPYFXF.M1 with a red moving average line. Below the chart is a sub-window for 'myBalance 9159,6300'. On the left, there is a '通貨ペア一覧' (Currency Pair List) and a 'ナビゲーター' (Navigator) window. At the bottom, the 'Expert Advisor' configuration window for 'myTest_SMA_03' is open, showing settings for the currency pair, model, and dates. The '日付と時間を使用' (Use Date and Time) checkbox is checked, and the '終了日' (End Date) is set to 2011.12.09.

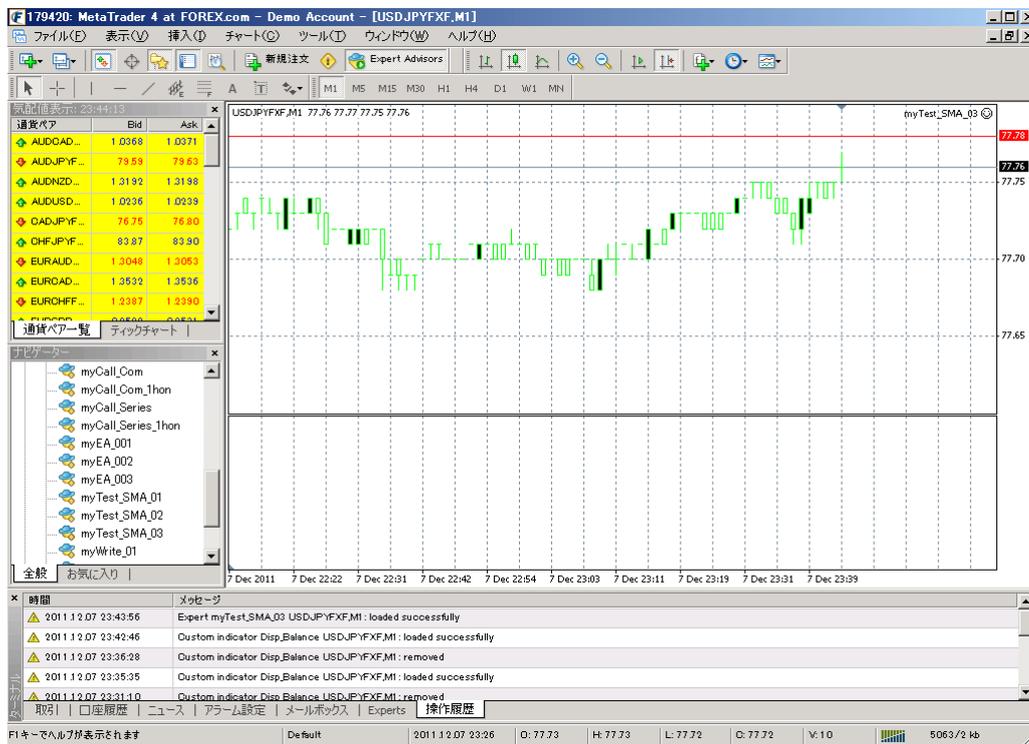


※このとき、設定されているインディケータをチェックすると下記のようになります。

The screenshot shows the '表示中の罫線分析ツール: USDJPYFXF.M1 (visual)' dialog box. It lists the active indicators: 'メイン・チャート' (Main Chart) with 'Moving Average' and 'サブウィンドウ 1' (Subwindow 1) with 'Disp_Balance'. There are buttons for '編集(E)' (Edit), '削除(D)' (Delete), and '閉じる(C)' (Close).

2. リアルチャート (デモ・モード) 上での表示

(1) 資産表示インディケータ「Disp_Balance.mq4」を、チャート上にセットする

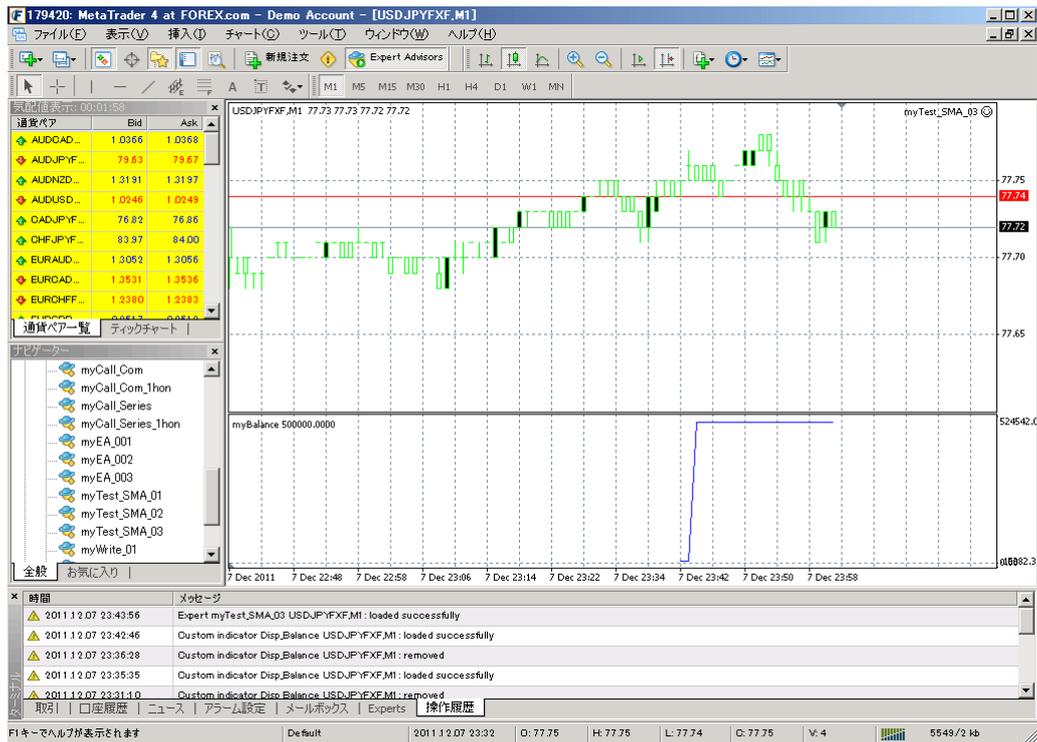


※この段階では、E Aを設定していないので、インディケータは何も表示しません。

※念のため、表示中のインディケータは下記で確認



(2) EA「myTest_SMA_03.mq4」をStrategy Testerにセットする



※資産カーブが初期化されて、すぐに表示を始めます。

⇒ EAの動作(実行)に従って、資産の変化(カーブ)が表示されていきます



※本稿で使用したEAは検討し始めたばかりのもので、資産カーブは右下がりになります。(改良できるところで報告するつもりのでEAです、ご容赦ください)
本当は、カッコよく、右肩上がりの資産カーブを表示したかったのですが、今回は間に合わず！

3. MQL4 コード内容一覧

```

(1) 動作確認用ストラテジー<EA>      「myTest_SMA_03.mq4」
//-----
//          myTest_SMA_03.mq4
//-----
#define myMAGIC  20111207
//
extern double MovingPeriod          = 12;
extern double MovingShift           = 6;
extern double Lots                   = 1;    // Regular
//-----
int init() {
    return(0);
}
//-----
int deinit() {
    return(0);
}
//-----
// オープンと待機中ポジションのチェック
//
int CheckCurrentOrders(string symbol)
{
    int buys=0, sells=0;
//----
    for(int i=0;i<OrdersTotal();i++)
    {
        if(OrderSelect(i, SELECT_BY_POS, MODE_TRADES)==false) break;
        if(OrderSymbol()==Symbol() && OrderMagicNumber()==myMAGIC)
        {
            if(OrderType()==OP_BUY)  buys++;
            if(OrderType()==OP_SELL) sells++;
        }
    }
//---- ポジション数を返す
    if(buys>0) return(buys);
    else      return(-sells);
}

//-----
// ポジションを開く (作成)
//
void Open_position()
{
    double ma;
    int    res;
//---- Open 値での動作用処理
//    if (Volume[0]>1) return;
//---- 移動平均値
    ma=iMA(NULL, 0, MovingPeriod, MovingShift, MODE_SMA, PRICE_CLOSE, 0);
//---- 売り
    if(Open[1]>ma && Close[1]<ma)
    {
        res=OrderSend(Symbol(), OP_SELL, Lots, Bid, 3, 0, 0, "", myMAGIC, 0, Red);
        return;
    }
}

```

```

//---- 買い
if(Open[1]<ma && Close[1]>ma)
{
    res=OrderSend(Symbol(), OP_BUY, Lots, Ask, 3, 0, 0, "", myMAGIC, 0, Blue);
    return;
}
//----

//-----
// ポジションを閉じる
//
void Close_position()
{
    double ma;
//---- Open 値での動作用処理
// if(Volume[0]>1) return;
//---- 移動平均値
ma=iMA(NULL, 0, MovingPeriod, MovingShift, MODE_SMA, PRICE_CLOSE, 0);
//----
for(int i=0;i<OrdersTotal();i++)
{
    if(OrderSelect(i, SELECT_BY_POS, MODE_TRADES)==false) break;
    if(OrderMagicNumber()!=myMAGIC || OrderSymbol()!=Symbol()) continue;
//---- オダー・タイプの確認
    if(OrderType()==OP_BUY)
    {
        if(Open[1]>ma && Close[1]<ma) OrderClose(OrderTicket(), Lots, Bid, 3, White);
        break;
    }
    if(OrderType()==OP_SELL)
    {
        if(Open[1]<ma && Close[1]>ma) OrderClose(OrderTicket(), Lots, Ask, 3, White);
        break;
    }
}
//----
}

//=====
//
void start()
{
//---- 資産データをグローバル変数に格納 ----
GlobalVariableSet("myBalance", AccountBalance());

//---- 足数とトレード可否をチェック
if(Bars<100 || IsTradeAllowed()==false) return;

//---- 待機中ポジションのチェックと、ポジション処理
if(CheckCurrentOrders(Symbol())==0) Open_position();
else Close_position();
//
}
//-----

```

(2) 資産カーブ表示用<インディケータ> 「Disp_Balance.mq4」

```
//
//  myBalance.mq4
#property indicator_separate_window
#property indicator_buffers 1
#property indicator_color1 Blue
//
double balance[];
//
int init()
{
    IndicatorShortName("myBalance");
    SetIndexStyle(0, DRAW_LINE);
    SetIndexBuffer(0, balance);
    SetIndexLabel(0, "Balance");
}
//
int deinit()
{
    return(0);
}
//-----
int start()
{
    //----
    balance[0]=GlobalVariableGet("myBalance");
    //----
    return(0);
}
//-----
```

※解説

< E A >

資産カーブを描画するために必要な記述は、下記の「青書」部分（一行）のみです。

```
GlobalVariableSet("myBalance", AccountBalance());
```

<インディケータ>

ポイントは、下記の「一行」のみです。

```
balance[0]=GlobalVariableGet("myBalance");
```

以 上