



IBI-Square Stocks

特別無料レポート

# 「自動売買システムを作ってみよう！」 その2

遊んでいても、仕事していても、寝ていても、パソコンが自動で売買を行ってくれる夢のシステム、自動売買システム。そんな自動売買システムを自分で作ってしまおうという無料レポートの第2弾です。

\*\*\*\*\*

～IBI-Square Stocks～ 代表：イビグチ ヤスヒロ

URL : <http://www.ibi-square.jp/> Mail : [stock@ibi-square.jp](mailto:stock@ibi-square.jp)

★『あつ、この銘柄上がりそう！！無料版』 <http://www.mag2.com/m/0000170473.html>

\*\*\*\*\*

Copyrights 2006 IBI-Square Stocks All Right Reserved.

無断複製・コピー・再配布等を禁止する。

## ■はじめに・・・

----- この無料レポートの趣旨 -----

このレポートは、データの読み込み、売買の判断（株の場合銘柄の選択も）、実際の注文、利益確定や手仕舞いの判断、実際の手仕舞い注文まで、すべて自動で行ってくれる自動売買システムをどのように構築していくか、私の経験を元に、そのために必要なもの、必要な知識を解説する特別無料レポートです。

巻末には私が構築した自動売買システム（相場からするとかなり安価）の紹介もありますのでそちらもご覧頂ければ幸いです。

## ■自動売買システムとは？

現在、自動売買システムというと、二つの意味合いがあるかと思います。

ひとつは、逆指値や IFD 注文を用いて、仕掛けと同時に手仕舞いを想定した注文も行い、自分で描いたシナリオに則って売買を行う方法を指したものです。この方法は、手仕舞いの部分を自動で行うために自動売買と呼ばれるのですが、実際の売買の判断や、仕掛け、注文を手作業で行わなくてはならず、完全な自動売買とはいえません。誤解なさないでほしいのは、先述の通り手仕舞いの部分に関しては自動ですので、このような方法を自動売買と呼称するのはわかりますし、この方法を批判しているわけではありません。

もうひとつは、このレポートで解説する自動売買です。こちらは、売買の判断、実際の注文、利益確定や損切りの判断、それに伴う実際の手仕舞いなど、すべて自動で行うシステムで、その間人間の判断、手作業でのパソコンの操作等を必要としないシステムです。

言い換えれば、そのシステムを起動させていれば、あなたが何かをしなくてもあなたに代わってパソコンが売買を行ってくれるというものです。

このレポートでは後者について、そのシステムの構築に必要な知識やツールの解説を行っていきます。

## ■ その1のおさらい

拙著「自動売買システムを作ってみよう！」では、銘柄の選択が必要なく比較的製作が簡単なFXの自動売買システムを構築する上で必要なモノ（データ、ツールなど）とそれらの処理について述べました。

未読の方は↓

[http://mag-zou.com/report\\_get.php?id=m0000170473\\_10&n=m0000170473](http://mag-zou.com/report_get.php?id=m0000170473_10&n=m0000170473)

ここで簡単におさらいしてみると、自動売買システムの構築に必要なモノというのは大きく分けて下記の4つになります。

- A) 為替のデータ（その提供元）
- B) 売買エンジン
- C) FX取引会社の提供する売買ツール
- D) 売買エンジンで判断した売買サインを実際に注文するためのツール

ここでいう売買エンジンとは、売買の判断を行うための条件、売買法のひとまとまりのことだとお考えください。

次に、上記の4つを元に各処理を行い、自動売買を行う手順としては、①為替データの読み込み、②データの書き込み、③テクニカル指標の計算・書き込み、④売買の判断、⑤売買の判断によって売買サインが出た場合は注文実行、と言う5つ処理になります。

当レポート「自動売買システムを作ってみよう！その2」では、上記4つモノの具体例、入手等と、5つの処理のうち①と②について具体的なコードと方法を掲載しています。

## ■ 構築の準備

自動売買システムの構築の準備として、まずは4つのモノが必要となります。それ以前に、ここでどのような自動売買システムを構築するか、その全体像を把握しておきましょう。

まず、先述のとおり、これから構築する自動売買システムは銘柄選択の必要がないFXを考えます。売買法としては、単純に移動平均線の短期と長期のクロスとします。(もちろん、それ以外でも可能ですが、製作手順の簡潔化と難易度を考えてです)

時間足は便宜上5分足として、移動平均線のパラメータは5本、25本にしたいと思います。

自動売買システムですので、断続的にデータを読み込む必要がありますが、その読み込み間隔は15秒とし、1分間に4回の読み込みを行います。もちろん、15秒の間には読み込めないデータも存在しますが、経験上5分足程度では大きな問題はありません。

システムの大元として、エクセルを使用し、処理はVBAにて行います。VBAを全く知らないという方は少々難儀かもしれませんが、その辺は初心者向けのVBAの書籍等もありますので、基本的な部分は省かせていただきます。

全体像としては、エクセルのシートに、為替データの4本値と日時を5分足で書き込み、それを元に2本の移動平均線を計算し記入します。

その後、ゴールドクロス、デッドクロスが発生した場合には、注文を行うというシステムです。当レポートは為替データの読み込みと書き込みまでを解説しますが、その後のテクニカル指標の計算、書き込み、クロス判断、注文に関しては、今後もその3、その4と続編にて解説していきたいと思います。

では、全体像がわかったところで準備です。

まず、エクセルが必要になります。おもちでない場合はご用意いただくしかないのですが、その場合はできればエクセル 2002 以上をご用意ください。レポート内の解説では 2002 での解説となります。2007 をお使いの場合は、エクセルのツールバーなどが大きく変更されていますので、その都度置き換えられてください。

次に、為替データをエクセルに読み込める形でデータを提供してくれるツールが必要です。

いくつかの取引会社が提供していますが、ここでは便宜上 FX オンラインを使用させていただきます。FX オンラインのデモ口座ならば即日解説が可能ですし、データの読み込み等も無料で無期限可能です。ということで、FX オンラインのデモ口座を開設されてください。

FX オンライン→ <https://www.fxonline.co.jp/system/demo.html>

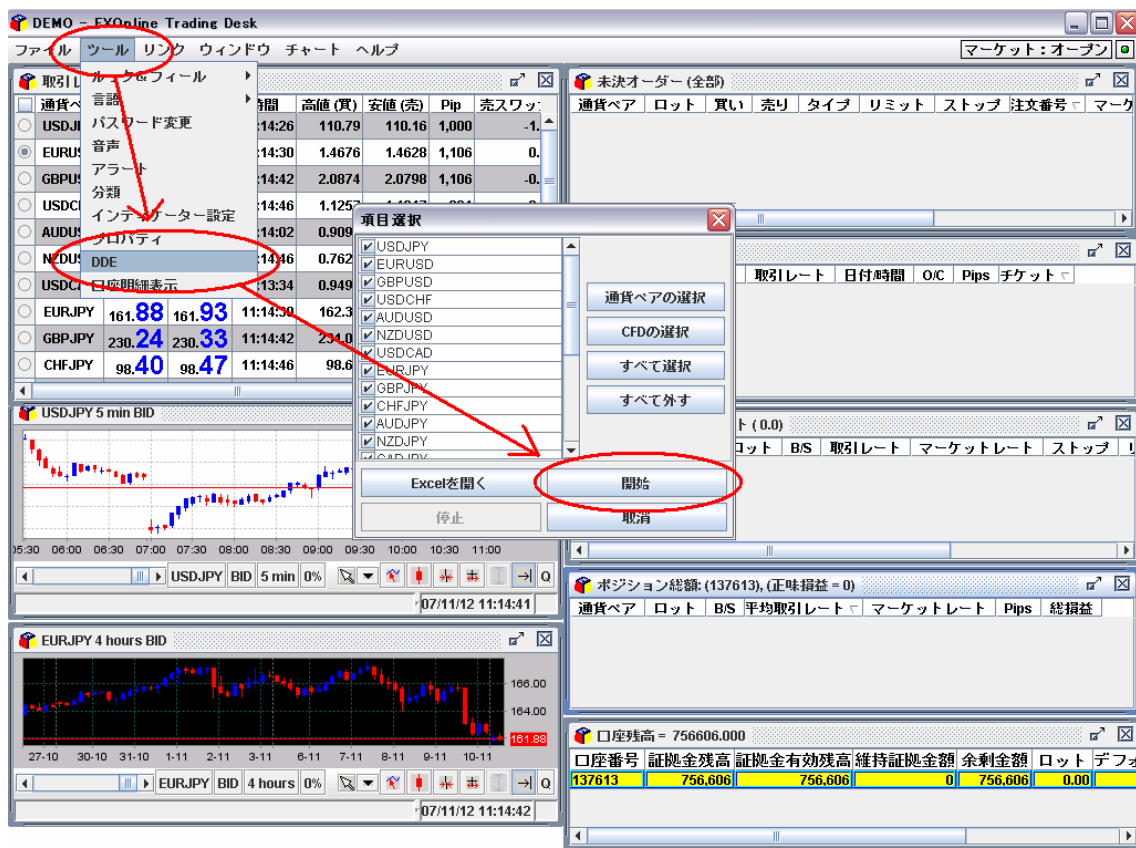
デモ口座申込後、メールが送られてきますので、指示に従ってシステムをダウンロードされてください。

当レポートまでの内容でしたら、上記二つがあれば可能です。実際に注文を行う部分などまで行くと、上記以外に注文を行う口座と注文を行うためのツールが必要となりますが、とりあえずはここまでで結構です。

## ■ 自動売買システムを作ってみよう！

では、準備が整ったところで早速作成に入っていきます。

まずは、FX オンラインのシステムにログインし、DDE というものに接続していただきます。この DDE がエクセルにデータを提供してくれるスグレモノ？です。方法は、FX オンラインの画面の上部にあるツールより、DDE を選択し、「開始」をクリックします。



これでエクセルにデータを読み込む準備が整いました。

次に、エクセルを開きます。

エクセルを開いたら、まずは DDE より FX のデータを読み込んでみましょう。方法は、エクセルのセルに読み込むための式を記入すれば良いだけです。

FX オンラインのデモ口座の DDE の読み込み式は下記のような形式になりますので、そちらをセル A1 に入力されてみてください。

```
=myFXDMO|rates!BidUSDJPY
```

上記式は US ドル/日本円の Bid レート（売値）を表示するための読込式となります。読込式の最後の 6 文字がそれぞれの通貨ペアを表しますので、ポンド/円を読み込またい場合は、「=myFXDEMO|rates!BidGBPJPY」となります。

次に、A2 に Ask レート（買値）を表示させてみましょう。Ask レートの読込式は

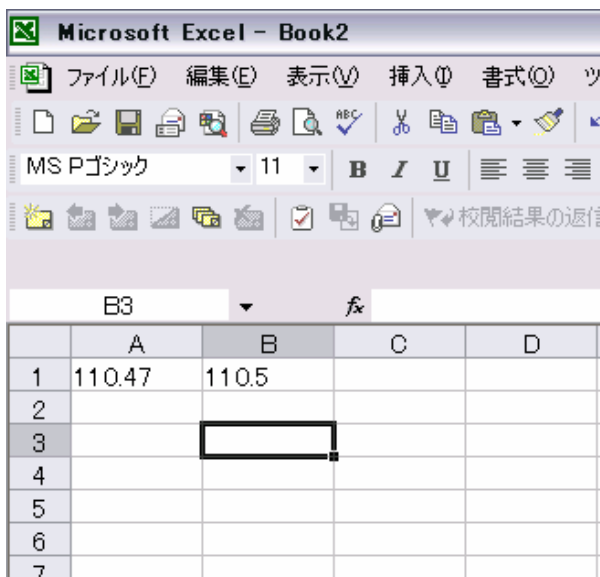
```
=myFXDMO|rates!AskUSDJPY
```

となります。

こちらも Bid レートと同様最後の 6 文字が通貨ペアとなりますので、先に入力した Bid レートのペアと同じペアの Ask レートを表示させてください。

ただ、Ask レートは便宜上表示させるだけで、テクニカル指標の計算等には Bid レートを使用します。

今の段階で、下図のような状態かと思えます。



The screenshot shows a Microsoft Excel window titled 'Microsoft Excel - Book2'. The spreadsheet has columns A, B, C, and D, and rows 1 through 7. Cell A1 contains the value '110.47' and cell B1 contains '110.5'. Cell B3 is currently selected and is empty. The formula bar above the spreadsheet is empty.

	A	B	C	D
1	110.47	110.5		
2				
3				
4				
5				
6				
7				

このとき、セル A1、B1 に表示されている内容が「#N/A」の場合は、DDE 側でレートの更新があり次第、レートが表示されますので問題ございません。

読み込み式を入力した際に、「外部データを更新できません」といった旨のメッセージが出てくる場合は、FX オンラインのシステムが正常に起動しているか、DDE に接続されているかをお確かめ下さい。その場合の式の内容は「#VALUE」や「#REF」となります。

では、データの読み込みができたところで、これから書き込みの部分に入って行くのですが、その前にエクセルの体裁を整えておきましょう。

まず、現在は前頁の画像の状態かと思いますが、1行目に行を挿入して、レートが表示されているセルを2行目に移します。同様に、1列目に列を挿入して、売値がセルB2に来るようにしてください。

そして1行目にそれぞれ売値、買値と見出しをつけておきましょう。また、セルの表示位置をセンターにし、お好みに合わせて文字やセルの大きさ、色、罫線などを調整されてください。(下図)

	A	B	C	D	E	F
1		売値	買値			
2		109.63	109.6			
3						
4						
5						

では、次に4本値を書き込む部分を作りますので、5行目の2列目から、始値、高値、安値、終値、日時と見出しをつけておきます。5行目の1列目には「#」といれ、1列目には行番号を入力するようにします。

	A	B	C	D	E	F	G
1		売値	買値				
2		109.57	109.54				
3							
4							
5	#	始値	高値	安値	終値	日時	
6							
7							
8							

最後に、時系列で 4 本値を書き込んでいくために、現在の行数や次の行に移るタイミングを判断させるためにいくつか数値の入力が必要となりますので、そちらを書き込んでおく場所を作りましょう。

どこでもいいのですが、便宜上セル E2 に行数、F2 に次の行へ移るタイミングを書き込むようにします。

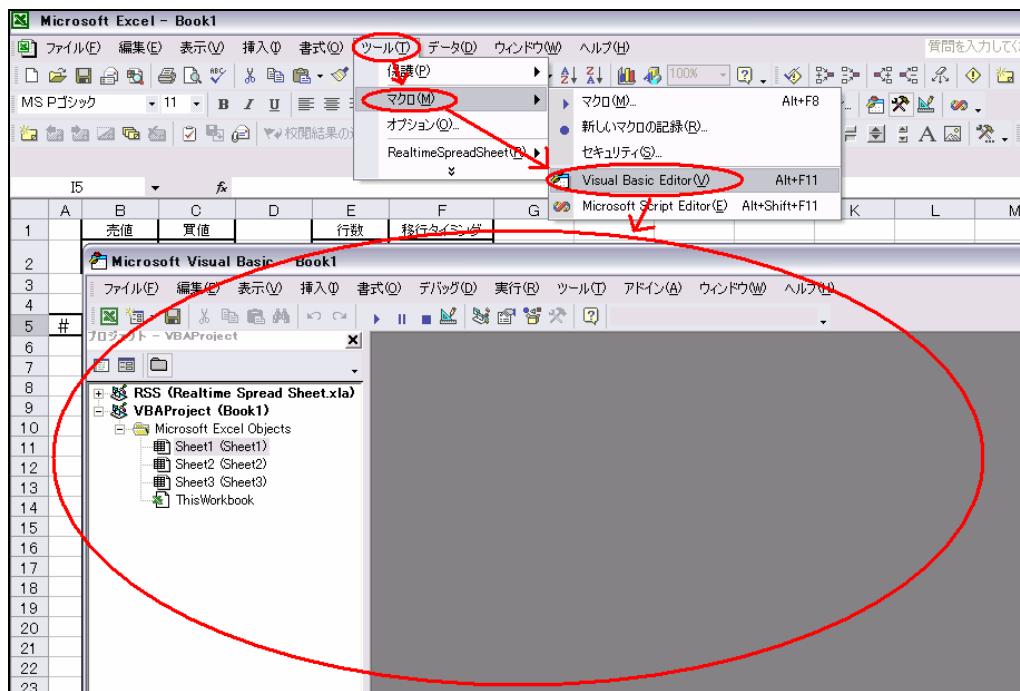
それぞれの 1 行目に見出しを付け、E2 には現在の行数である 6（4 本値の見出しを 5 行目につけたため、現在の行は 6 行目となります）を、F2 は 0 を入力しておきます。

罫線などを調整して、下図のような感じです。

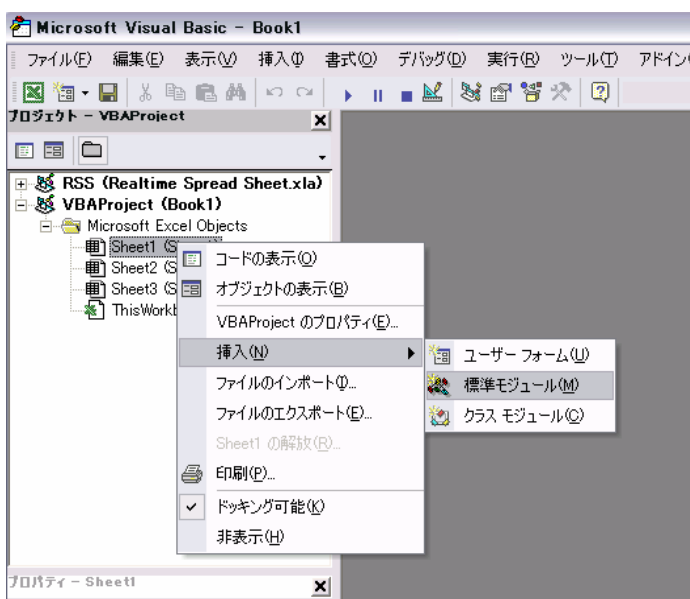
	A	B	C	D	E	F	G
1		売値	買値		行数	移行タイミング	
2		109.68	109.65		6	0	
3							
4							
5	#	始値	高値	安値	終値	日時	
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							

ここまで完了すると、いよいよ VBA のコードの記入に入ります。

まずは、VBA を起動します。エクセルの「ツール」より「マクロ」にカーソルを合わせ、「Visual Basic Editor」を選択します。

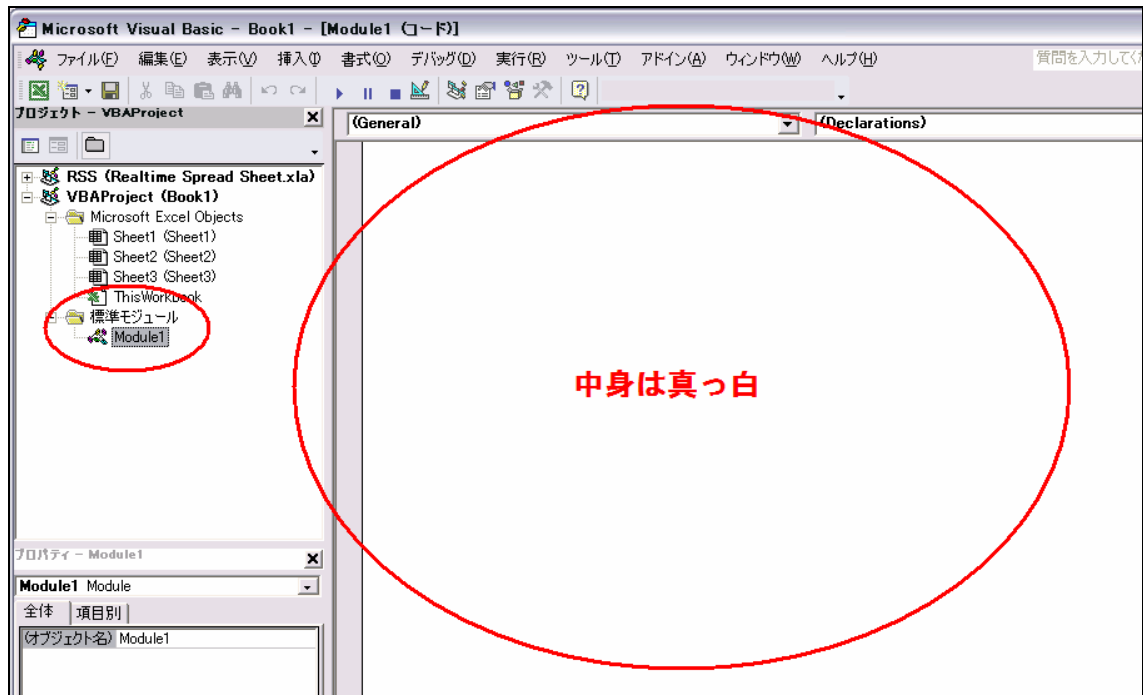


VBA を表示させたら、まずは標準モジュールを追加します。



画面左側にあるプロジェクトウィンドウの任意の場所で右クリックをし、「挿入」、「標準モジュール」の順でクリックします。

すると標準モジュールが追加され内容が表示されます。まだコードなどは何も書いていないので、真っさらな状態です。



この右側の真っ白の部分にコードを書いていきます。

私も VBA を誰かに教えて貰ったわけではなく、ほとんど独学ですのでプロの方や知識の豊富な方に見れば「そこはこうだよ！」とか「こうした方がいいのでは？」「この表現はおかしい！」と言われるかもしれませんが、その辺はご了承ください。

コードですが、ここで書きこむコードは 30 分足の作成の部分です。大きく分けると 2 つの動作を制御させます。

一つ目は、最初の更新で始値、高値、安値、終値を書き込み、次の更新では現在の値と記入されている高値を比較し、現在の値が高ければ高値を更新、同様に書き込まれている安値と現在の値を比較し、現在の値が低ければ更新、最後に現在の値を現在の終値として書き込むという作業です。

二つ目が 5 分足の 4 本値を書き込むため、現在の時間を判断して、分の数字が 00 分、05 分、10 分、…50 分、55 分だった場合に次の行に移るような制御です。

では早速まずは4本値の書き込み部分のコードを書いてきましょう。

まずは、右側の真っ白な部分の一番上に、「Sub Kakikomi ()」と入力してエンターキーをクリックします。すると「End Sub」が自動で挿入されます。

「Sub Kakikomi ()」と「End Sub」の間に行う処理を記述していきます。

最初に、現在のレート、現在の行数を読み込みます。

現在のレートはセル B2 に表示されている売値を使用しますので、変数“r\_now”に現在のレートを代入します。ただし、そのまま DDE で配信された値を読み込んでも、文字列として読み込んでしまいますので、数値に変換して読み込みます。数値に変換するには、Val を使用します。

```
r_now = Val (Cells (2, 2))
```

現在の行数は9ページの状態ですとセル E2 に表示されていますので、そちらの値を変数“num\_1”という変数に代入します。

```
Num_1 = Cells (2, 5)
```

※ VBA でのセルの参照の場合、列はエクセルに表示の A, B, C, …ではなく、それぞれ番号での指定となります。セルの指定は「Cells」の後に行番号、列番号の順で番号を記入します。たとえば、セル B3 は3行目のB列=2列目ですので、Cells (3, 2) となります。

現在までのコードの状態は下記の様になるかと思えます。

```
(General)

Sub Kakikomi()

    r_now = Val (Cells (2, 2))
    num_1 = Cells (2, 5)

End Sub
```

次に、その行での最初の更新の時は、まだデータが書き込まれていない状態で、始値も高値も安値も終値も現在のレートとなりますので、その部分のコードを記入します。

コードは、まず現在の行（行数は num\_1）の 2 列目、3 列目、4 列目、5 列目に現在のレート（r\_now）を書き込むコードです。

「現在の行の 2 列目」は「Cells(num\_1, 2)」ですので、そこに現在のレートを記入します。

```
Cells(num_1, 2) = r_now
```

同様に、3 列目、4 列目、5 列目にも r\_now を記入します。

```
Cells(num_1, 3) = r_now
```

```
Cells(num_1, 4) = r_now
```

```
Cells(num_1, 5) = r_now
```

これらの処理は現在の行に書き込みがない場合に行いますので、ここで条件分岐式を入力しておきます。

条件分岐は「If~Else~End If」で行いますが、現在の行に全く書き込みがない場合は、始値を記入するセルにレートが書き込まれていない状態ですので、条件は「その行の始値のセルが白紙ならば」という条件になります。

現在の行の始値のセルは「Cells(num\_1, 2)」で白紙は「""」で表します。これらを「If」に続けて条件として記入し、その中にその条件を満たした場合の処理を記入します。

既に処理は記入済みですので、上記の 4 行の上に

```
If Cells(num_1, 2) = "" Then
```

と記入します。

If を記入したら、それに対応する「End If」が必要ですので、処理の最後に「End If」を記入するのですが、その前に、条件を満たさなかった場合には、高値と現在値の比較、安値と現在値の比較、終値への現在値の記入の処理を行いますので、そちらを先に記入します。

条件を満たさなかった場合、つまり現在の行の始値が白紙ではなかった場合は、高値、安値の比較、終値の記入という処理が必要です。

まずは、現在記入されている高値と安値を読み込んでおきます。

現在の書き込まれている高値を「r\_high」、安値を「r\_low」に代入しておきます。

```
r_high = Cells(num_1, 3)
```

```
r_low = Cells(num_1, 4)
```

次に比較ですが、ここでも If を使用して比較します。現在のレートは r\_now に入っていますので、たとえば、現在値と書き込まれている高値の比較の条件は、「If r\_now > r\_high Then」となります。

同様に、安値の比較は「If r\_now < r\_low Then」となります。

それぞれの条件とそれぞれの処理を記入すると、高値、安値の比較、及びその書き込みは下記のコードになります。

```
If r_now > r_high Then  
    Cells(num_1, 3) = r_now  
End If  
If r_now < r_low Then  
    Cells(num_1, 4) = r_now  
End If
```

上記のコードは、「If r\_now > r\_high Then」で現在値と書き込まれている高値とを比較し、現在値の方が大きい場合は、現在値を高値の部分に書き込む処理を行うコード、及び、現在値と書き込まれている安値とを比較し、現在値の方が小さい場合は、現在値を安値の部分に書き込む処理を行うコードです。

そして、最後に現在値を終値の部分に書き込みます。この場合は条件分岐の必要はありませんので、直接書き込みます。

```
Cells(num_1, 5) = r_now
```

ここまでを記入すると、下記の一連のコードとなります。

(General)
Kakikomi

```

Sub Kakikomi()

r_now = Val(Cells(2, 2))
num_1 = Cells(2, 5)

If Cells(num_1, 2) = "" Then
    Cells(num_1, 2) = r_now
    Cells(num_1, 3) = r_now
    Cells(num_1, 4) = r_now
    Cells(num_1, 5) = r_now

    r_high = Cells(num_1, 3)
    r_low = Cells(num_1, 4)

    If r_now > r_high Then
        Cells(num_1, 3) = r_now
    End If
    If r_now < r_low Then
        Cells(num_1, 4) = r_now
    End If

    Cells(num_1, 5) = r_now

End Sub

```

ここで、2 ページ前に戻って、上記のコードには条件を満たした場合の処理と満たさなかった場合の処理が記入してあります。条件は、「その行の始値が白紙ならば」というものでした。上記コード内の「条件」の部分です。

この条件を満たしたら、始値～終値に現在値を記入し、満たさなかったら高値、安値の比較と終値への記入を行うコードを書いています。

では、満たした場合の処理と満たさなかった場合の処理の間に「Else」と記入し、それぞれの処理を別のプロセスにします。

最後に、If を使用したらそれに対応する End If が必要ですので、それを記入しましょう。

**(General)**

```

Sub Kakikomi()

r_now = Val(Cells(2, 2))
num_1 = Cells(2, 5)

If Cells(num_1, 2) = "" Then

    Cells(num_1, 2) = r_now
    Cells(num_1, 3) = r_now
    Cells(num_1, 4) = r_now
    Cells(num_1, 5) = r_now

Else ← 処理と処理の間に
      Elseを記入

    r_high = Cells(num_1, 3)
    r_low = Cells(num_1, 4)

    If r_now > r_high Then
    Cells(num_1, 3) = r_now
    End If
    If r_now < r_low Then
    Cells(num_1, 4) = r_now
    End If

    Cells(num_1, 5) = r_now

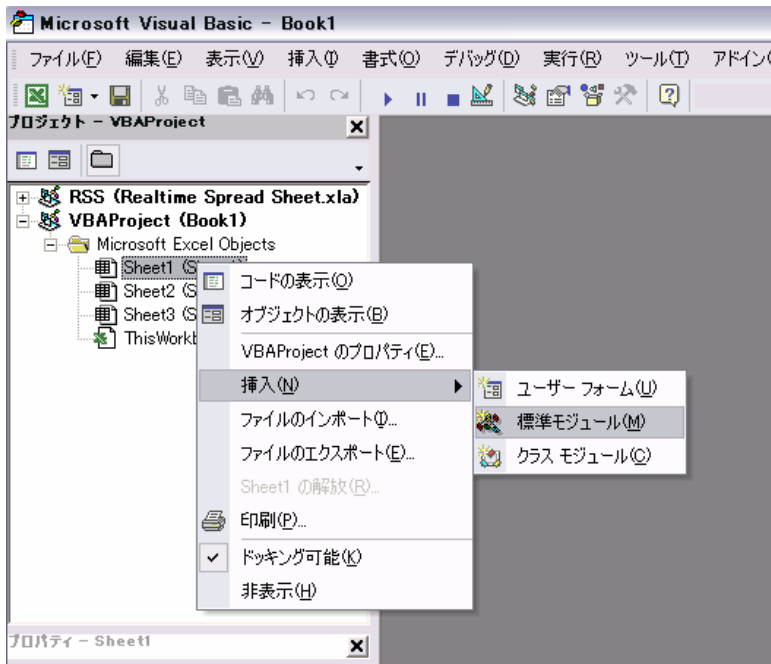
End If ← If に対応するEnd Ifを記入

End Sub
  
```

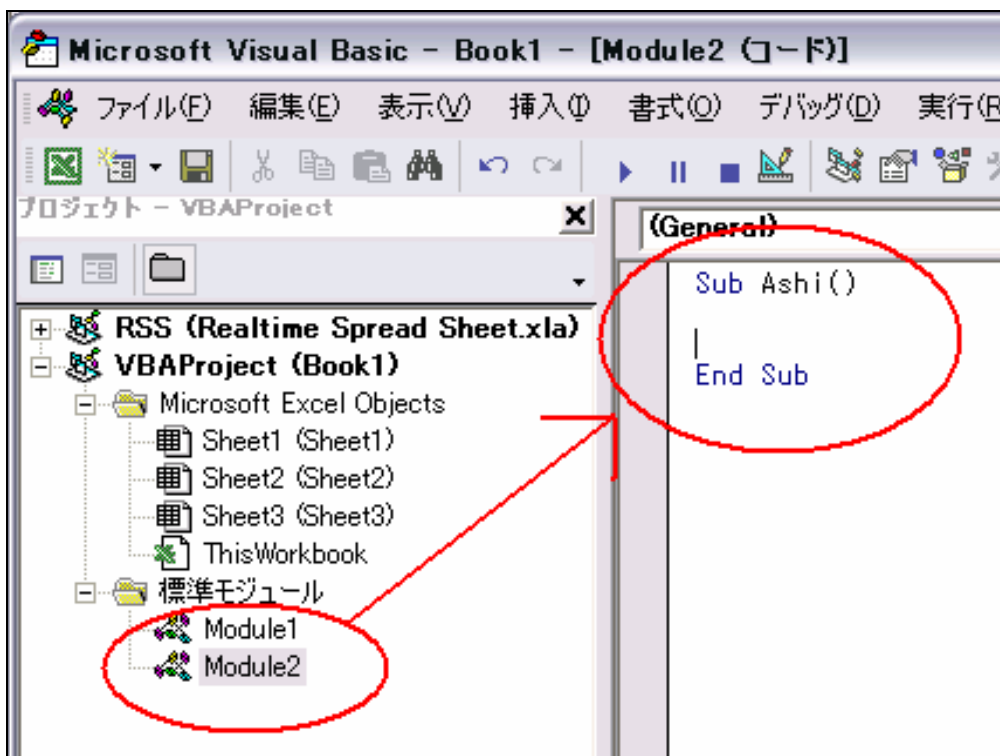
ここまでの、4本値の記入は完了です。

次は、時間を判断して行を下へ移行していく処理を行います。

まずは新しい標準モジュールを追加します。



追加したら、今回は「Sub Ashi ()」と記入して、下図の状態にしてください。



記入が完了したら、時間の制御を行うために時間をシートに書き込むコードを記入します。

書き込みの前に、変数に時間を代入します。

時間は変数 `time_n` に文字列で代入し、あとから判断を行いやすくします。

文字列で代入する場合は、常に一定の形式で代入できるように書式を指定します。コードは、「Format(Now, "yyyy/mm/dd hh:mm:ss")」になります。これは、現在値を「年/月/日 時:分:秒」という形式で代入するコードです。

これを記入します。

次に、5分足の描画ですので、00分、05分、10分、15分、20分、25分、30分、35分、40分、45分、50分、55分に行の更新が行われますので、それらを条件分岐にて指定します。

シートには現在の行数を表示するようになっていますので、上記の時間になったら、その行数をひとつ増やせば良いわけです。

条件分岐は先述の `If` を使用しますが、今回は条件が複数あり、条件のうちいずれかを満たせば条件クリアとなる `OR` 条件となります。`OR` 条件は各条件を `OR` で接続すれば `OK` です。

上記の分単位の時間の判断については、すでに変数 `time_n` に文字列として時間が入力されていますので、その“分”の部分を抽出し、それを元に判断が可能です。抽出は、変数 `time_n` (文字列) の `x` 文字目から `y` 文字を抽出する「`Mid`」という関数を使用します。

現在の時間 `time_n` の“分”の内容は、`time_n` の 15 文字目から 2 文字分にありますので、それを抽出します。それを変数 `min_n` に代入します。

コードは `min_n = Mid(time_n, 15, 2)` となります。

ここまでのコードは下記の通りです。

<b>(General)</b>	
	<pre> Sub Ashi()  time_n = Format(Now, "yyyy/mm/dd hh:mm:ss")  min_n = Mid(time_n, 15, 2)  End Sub           </pre>

次に、先述の“分”を判断する条件分岐式を記入します。

OR 条件は各条件を OR で結べば良いので、変数 min\_n が“00”や“05”・・・であったらという条件を順に記入します。

If min\_n = “00” OR min\_n = “05” OR …と記入していくのですが、コードが横に長くなるので、改行を適宜挿入します。コードの途中で改行を入れる場合は半角スペースに「\_」を記入して改行します。

```

If min_n = "00" Or min_n = "05" _
Or min_n = "10" Or min_n = "15" _
Or min_n = "20" Or min_n = "25" _
Or min_n = "30" Or min_n = "35" _
Or min_n = "40" Or min_n = "45" _
Or min_n = "50" Or min_n = "55" Then

```

上記の条件を満たしたときに現在の行数が一つ増えるので、条件を満たしたときのコードは下記の様になります。

```

num_1 = Cells(2, 5)
Cells(2, 5) = num_1 + 1

```

上記のコードは Cells(2, 5) の値を変数 num\_1 に代入し、それに 1 加えた値を Cells(2, 5) に書き込みます。

併せて、If を使用したら、それに対応する End If が必要ですので、それも記入します。

```

(General)

Sub Ashi()

time_n = Format(Now, "yyyy/mm/dd hh:mm:ss")
min_n = Mid(time_n, 15, 2)

    If min_n = "00" Or min_n = "05" -
    Or min_n = "10" Or min_n = "15" -
    Or min_n = "20" Or min_n = "25" -
    Or min_n = "30" Or min_n = "35" -
    Or min_n = "40" Or min_n = "45" -
    Or min_n = "50" Or min_n = "55" Then

        num_1 = Cells(2, 5)
        Cells(2, 5) = num_1 + 1

    End If

End Sub
  
```

これで、更新時の“分”が5分足の始まりの時間であった場合は行数を増加させることができます。

しかし、上記のコードでは正常に動きません。

というのも、最初に全体像を説明した際に掲載したとおり、更新は15秒毎に行う予定です。15秒毎に行うと、たとえば00分での更新は4回ありますので、その都度行を更新してしまうこととなります。その処理を書き込まないといけません。

そこで、シートに作成してもらった移行タイミングを使用します。

現在、移行タイミングが記入してあるセルF2には「0」が記入してありますが、そこが1の場合にのみ、上記のコードを稼働させるようにします。

移行タイミングが0か1かは、「行を更新したら→0」、「現在の“分”が00分、05分、…、50分、55分でなかったら→1」という制御を行えば、行を1度更新すると、“分”の値が00分、05分、…、50分、55分ではない時間帯になるまで0の状態にできます。0の状態では行の移行は行いませんし、“分”の値が00分、05分、…、50分、55分でない場合も移行を行いませんので、通常の5分足の様に、5分に1度だけ行を更新することになります。

コードでは、まずここまでに書いている条件分岐式（“分”の判断部分）すべてを、「移行タイミングが1ならば」実行するように条件分岐で囲みます。

つまり、セルF2の値が1ならば実行しますので、コードは

```
If Cells(2, 6) = 1 then
```

となります。

次に、1度更新したら、移行タイミングは0にしなくてはなりませんので、それを行の更新直後に記入するようにします。

```

Sub Ashi()
  time_n = Format(Now, "yyyy/mm/dd hh:mm:ss")
  min_n = Mid(time_n, 15, 2)
  If Cells(2, 6) = 1 Then
    If min_n = "00" Or min_n = "05" -
    Or min_n = "10" Or min_n = "15" -
    Or min_n = "20" Or min_n = "25" -
    Or min_n = "30" Or min_n = "35" -
    Or min_n = "40" Or min_n = "45" -
    Or min_n = "50" Or min_n = "55" Then
      num_1 = Cells(2, 5)
      Cells(2, 5) = num_1 + 1
      Cells(2, 6) = 0
    End If
  End If
End Sub
  
```

そして、“分”の値が 00 分、05 分、…50 分、55 分でなかった場合に移行タイミングを 1 に戻しますので、そのコードを書きます。

条件式「〇〇が××でなかったら」は、〇〇 <> ××という式になりますので、それを最後に書き加えます。このとき、この条件式は毎回判断を行いますので、「移行タイミングが 1 ならば」という条件分岐の外に書きます。また、今回は 00 分、05 分、…50 分、55 分のいずれでもなかった場合に条件クリアとなりますので、条件は AND 式となります。

コードは下記の通り。

```
If min_n <> "00" And min_n <> "05" _  
And min_n <> "10" And min_n <> "15" _  
And min_n <> "20" And min_n <> "25" _  
And min_n <> "30" And min_n <> "35" _  
And min_n <> "40" And min_n <> "45" _  
And min_n <> "50" And min_n <> "55" Then
```

条件をクリアした際の処理は、移行タイミングを 1 にしますので、  
Cells(2, 6) = 1  
となります。

If に対応する End If をお忘れなく。

では、最後に日時の書き込みと、最初に作成した「Sub Kakikomi ()」への接続を行います。

日時の書き込みは、これまでの処理にて現在の行数が変わっている（更新されている）場合もありますので、もう一度 num\_1 の値を読み込みます。

```
num_1 = Cells(2, 5)
```

そして、最初に読み込んだ time\_n を日時の列に書き込みます。

```
Cells(num_1, 6) = time_n
```

ここまでが完了したら、先ほど作成した Sub Kakikomi () を呼び出します。他のマクロを呼び出す際は Call を使用しますので、下記の様なコードとなります。

```
Call Kakikomi
```

これらを Sub ashi () の最後に追加して、一連の動作のコードは完了となります。

```

(Sub) Ashi
time_n = Format(Now, "yyyy/mm/dd hh:mm:ss")
min_n = Mid(time_n, 15, 2)
If Cells(2, 6) = 1 Then
    If min_n = "00" Or min_n = "05" -
    Or min_n = "10" Or min_n = "15" -
    Or min_n = "20" Or min_n = "25" -
    Or min_n = "30" Or min_n = "35" -
    Or min_n = "40" Or min_n = "45" -
    Or min_n = "50" Or min_n = "55" Then

        num_1 = Cells(2, 5)
        Cells(2, 5) = num_1 + 1

        Cells(2, 6) = 0

    End If
End If

If min_n <> "00" And min_n <> "05" -
And min_n <> "10" And min_n <> "15" -
And min_n <> "20" And min_n <> "25" -
And min_n <> "30" And min_n <> "35" -
And min_n <> "40" And min_n <> "45" -
And min_n <> "50" And min_n <> "55" Then

    Cells(2, 6) = 1
End If

num_1 = Cells(2, 5)
Cells(num_1, 6) = time_n

Call Kakikomi

End Sub
  
```

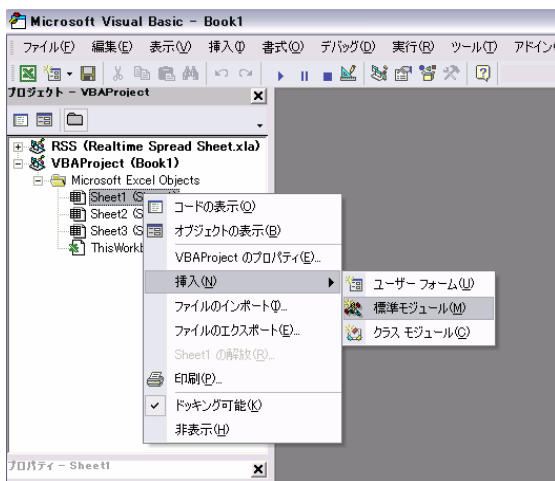
指定の"分"以外ならば  
 移行タイミングを1に  
 日時の書き込み  
 Kakikomi()の呼び出し

さて、ここまでで時間足書き込みの部分は完了です。

あと、今回のレポートで残っている部分は、これらのコードを 15 秒毎に動作させるコードのみとなりました。

こちらはタスクを使用しますので、少々難易度が高く、私自身もそれほど詳しく理解していませんので、コードの提示をして軽く解説するだけとさせていただきたいと思います。もちろん、稼働するには十分耐えうるものですのでご安心ください。

では、これまで通り新しい標準モジュールを追加してください。



モジュールを追加したら、今回は一番上の部分に「Dim mcolTask As Collection」と入力してください。

その後、Sub Tasc\_Bigin() とマクロのタイトルをつけ、コードを書いていきます。

```
(General)
Dim mcolTask As Collection

Sub Tasc_Bigin()

End Sub
```

まずは、開始時刻、終了時刻、更新間隔などのセッティング部分ですが、この部分に関しては別途テキストファイルを用意していますので、そちらからコピーされても結構です。

テキストファイル→ [http://www.ibi-square.jp/file/atm2\\_1.txt](http://www.ibi-square.jp/file/atm2_1.txt)

(標準モジュールを追加した後、ファイルの中身をすべてコピーしていただければ OK です。)

コードの内容を上から順に説明します。

まず、セッティングの部分として下記のようなコードがあるかと思えます。

```
' セッティング-----
dayBigin = "07/01/01"           ' 開始日付
datBigin = TimeValue("00:00:00") ' 開始時刻
dayEnd = "07/01/01"           ' 終了日付
datEnd = TimeValue("00:00:00")  ' 終了時刻
datInterval = TimeValue("00:00:15") ' 実行間隔(少なくとも数秒以上で)
datTimeout = TimeValue("00:02:00") ' 実行待機タイムアウト
blnJustTime = True             ' datInterval で丸めるか
strProcName = "Ashi"          ' 実行するマクロ名
int5 = TimeValue("00:00:05")   ' 設定した終了時間からのマックス時間
-----'
```

こちらは説明のある通りで、設定していただくのは、開始日付、開始時刻、終了日付、終了時刻、実行するマクロ名です。

当然ですが、終了日付+終了時刻よりも、開始日付+開始時刻の方が早い時間となります。実行するマクロは、今回は「Ashi」となりますので、すでにそちらが設定済みです。

実行間隔は全体像で解説したとおり 15 秒としています。

あとは、下記コードを書きいただければ稼働させることが可能です。

<以下、コード>

```
If mcolTask Is Nothing Then
' 日付シリアル値を加算
datBegin = datBegin + Application.WorksheetFunction.Text(dayBegin,
"yyyy/mm/dd")
datEnd = datEnd + Application.WorksheetFunction.Text(dayEnd,
"yyyy/mm/dd") - int5

If datEnd < datBegin Then
    MsgBox "開始日時が終了日時より遅くなっています。"
    Exit Sub
End If

' 現在時刻が既に終了時刻を過ぎている場合
If datEnd < Now() Then
    MsgBox "終了時刻を過ぎているため予約できません。", vbCritical, "終了"
    Exit Sub
End If

' 現在時刻が開始時刻を過ぎていれば補正
If datBegin <= Now() Then
' 開始時刻を datInterval で指定された値で丸めるか
If blnJustTime Then
    datBegin = Application.Floor(Now() + datInterval, datInterval)
Else
    datBegin = Now() + datInterval
End If
End If

' 初期化
Set mcolTask = New Collection

' メイン部分
For i = datBegin To datEnd Step datInterval
' 後から取り消せるようにコレクションに退避
mcolTask.Add CStr(i) & ", " & strProcName
```

```
' Application.OnTime で実行予約を行う
Application.OnTime EarliestTime:=i, _
                Procedure:=strProcName, _
                LatestTime:=i + datTimeout, _
                Schedule:=True
    Next i

Else
    MsgBox "既に実行中です", vbInformation
End If

Exit Sub
error:
    MsgBox "エラーが発生しました。設定された日時をご確認ください。"
End Sub
```

```

(デ) デバッグ(D) 実行(R) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H) 質問を入力して
67行, 1桁
general) Tasc_Bigin
Dim mcolTask As Collection

Sub Tasc_Bigin()
' セッティング-----
    dayBigin = "2007/11/14" ' 開始日付
    datBigin = TimeValue("09:22:00") ' 開始時刻
    dayEnd = "2007/11/14" ' 終了日付
    datEnd = TimeValue("10:00:00") ' 終了時刻
    datInterval = TimeValue("00:00:15") ' 実行間隔(少なくとも数秒以上で)
    datTimeout = TimeValue("00:02:00") ' 実行待機タイムアウト
    blnJustTime = True ' datInterval で丸めるか
    strProcName = "Ashi" ' 実行するマクロ名
    int5 = TimeValue("00:00:05") ' 設定した終了時間からマイナスする時間
' -----

If mcolTask Is Nothing Then
' 日付シリアル値を加算
    datBigin = datBigin + DateValue(dayBigin)
    datEnd = datEnd + DateValue(dayEnd) - int5

    If datEnd < datBigin Then
        MsgBox "開始日時が終了日時より遅くなっています。"
        Exit Sub
    End If

' 現在時刻が既に終了時刻を過ぎている場合
    If datEnd < Now() Then
        MsgBox "終了時刻を過ぎているため予約できません。", vbCritical, "終了"
        Exit Sub
    End If

' 現在時刻が開始時刻を過ぎていれば補正
    If datBigin <= Now() Then
        ' 開始時刻を datInterval で指定された値で丸めるか
        If blnJustTime Then
            datBigin = Application.Floor(Now() + datInterval, datInterval)
        Else
            datBigin = Now() + datInterval
        End If
    End If

' 初期化
    Set mcolTask = New Collection

' メイン部分
    For i = datBigin To datEnd Step datInterval
        ' 後から取り消せるようにコレクションに退避
        mcolTask.Add CStr(i) & "," & strProcName
        ' Application.OnTime で実行予約を行う
        Application.OnTime EarliestTime:=i, _
            Procedure:=strProcName, _
            LatestTime:=i + datTimeout, _
            Schedule:=True

    Next i

Else
    MsgBox "既に実行中です", vbInformation
End If

Exit Sub

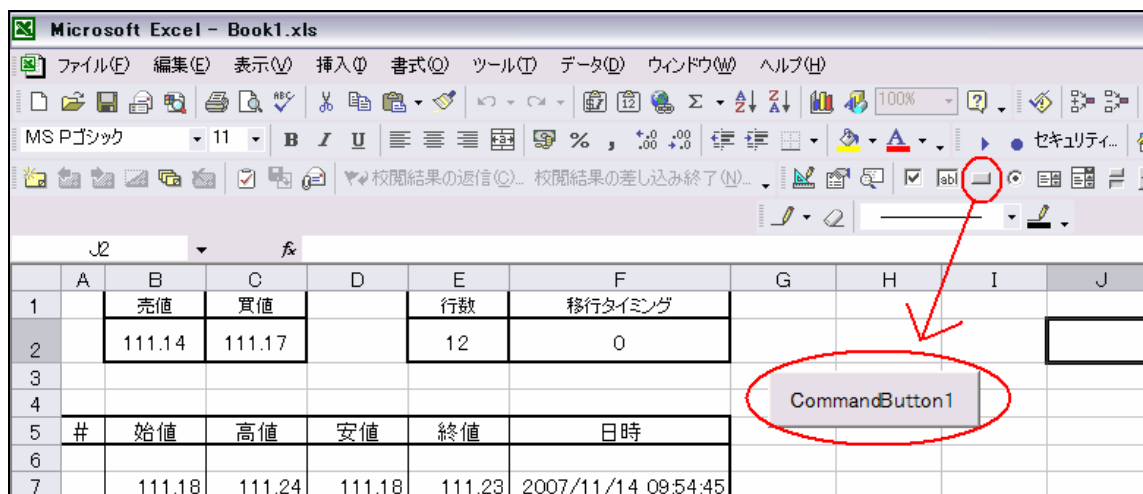
error:
    MsgBox "エラーが発生しました。設定された日時をご確認ください。"

End Sub
    
```

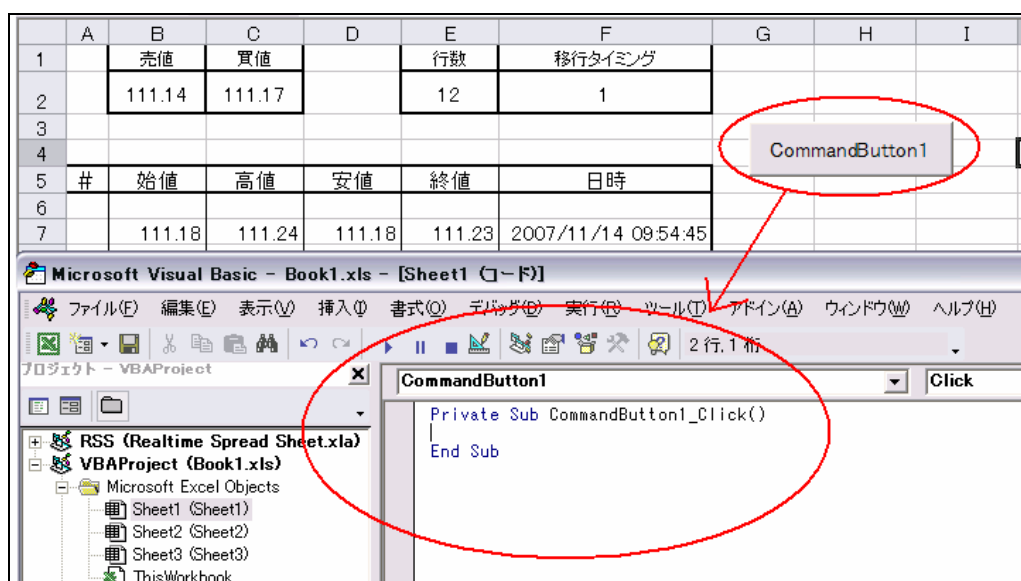
さて、ここまでで 15 秒毎の読み込み、行の移行、4 本値の書き込みと、当レポートでの内容は網羅しました。

では、シートに戻って、ボタンの配置と行番号の挿入をして終わりしたいと思います。

シートに戻って、ツールボックスのコントロールより、ボタンを配置しましょう。

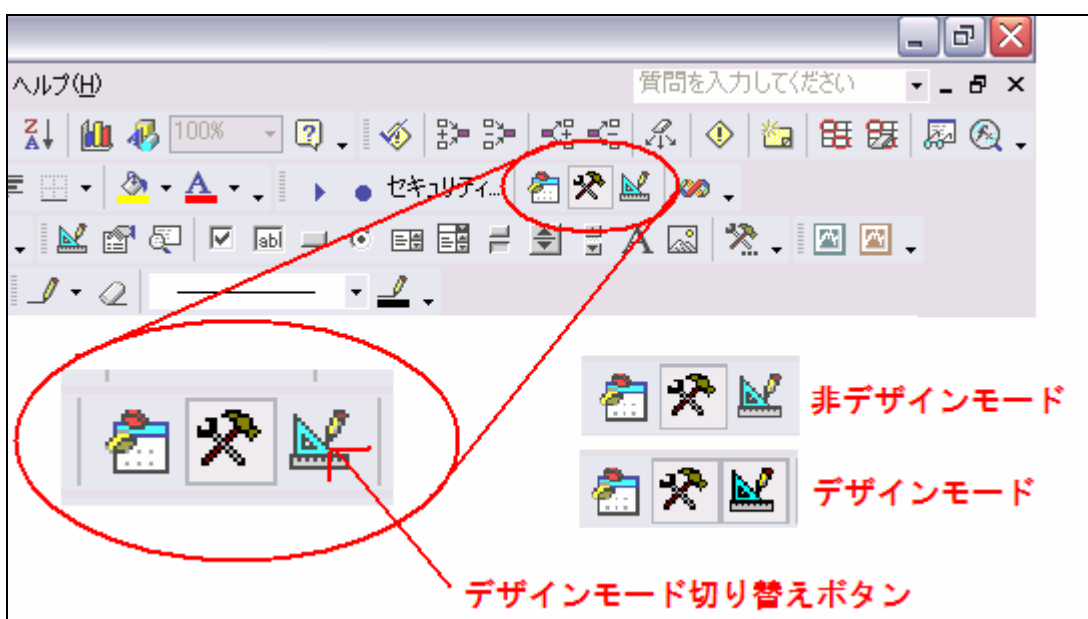


ボタンを配置したら、そのボタンをダブルクリックします。すると、下記のようなモジュール画面が表示されますので、そのコードに、「Call Tasc\_Bigin」と入力します。



```
CommandButton1
Private Sub CommandButton1_Click()
    Call Tasc_Bigin
End Sub
```

※ボタンをダブルクリックしても表示できない場合は、コントロールをデザインモードにしてお試しください。(下図)



この処理にて、シート上で今作成したボタンをクリックすると、コード内に設定した日時、時間に則って更新を開始します。

では、行番号を入れて（次回のレポート以降で使用します。）終わりしたいと思います。

シートの 1 列目に行番号を入れて、日時や時刻の設定を済ませ、ボタンをクリックしましょう。その際は、FX オンラインに接続され、DDE が正常に稼働しているかどうかをご確認の上、実行されてください。

尚、一度保存されたあと、再度起動される際には、「マクロを有効にするかどうか」、「外部データを更新するかどうか」のメッセージが表示されますが、その際はそれぞれ「有効にする」「更新する」を選択されてください。

行番号を入れて、時刻等設定後ボタンを押すと時間の経過とともに、自動で5分足の4本値を書き込んでいきます。

	A	B	C	D	E	F	G	H	I
1		売値	買値		行数	移行タイミング			
2		111.12	111.15		14	1			
3									
4									
5	#	始値	高値	安値	終値	日時			
6	1								
7	2	111.18	111.24	111.18	111.23	2007/11/14 09:54:45			
8	3	111.22	111.23	111.21	111.21	2007/11/14 09:59:45			
9	4	111.22	111.22	111.12	111.12	2007/11/14 10:04:45			
10	5	111.12	111.13	111.11	111.13	2007/11/14 10:09:45			
11	6	111.13	111.14	111.12	111.14	2007/11/14 10:14:45			
12	7	111.14	111.15	111.09	111.11	2007/11/14 10:19:45			
13	8	111.12	111.16	111.12	111.14	2007/11/14 10:24:45			
14	9	111.14	111.14	111.12	111.12	2007/11/14 10:29:30			
15	10								
16	11								

さて、ここまでが今回の内容です。ちょっと煩雑な部分もありましたが、データの読み込みまで行えていますでしょうか？

次回は移動平均線の書き込み部分と、そのクロスでの売買判断を考えていきたいと思います。

ここでは、読者の皆さんに作成していただく手順を、私の経験を基に解説してきました。今回はデータの読み込みから、時間足の書き込みまでの基本的な部分を開設してきましたが、ある程度の知識がたまり、次回以降開設予定の移

動平均線や、それを基にした売買など、基本的な処理を行えるようになると、他にもいろいろと実装させたい機能が出てくるものです。

私が作成した自動売買システムを、最初は基本的なものだけでした。売買法も移動平均線を使用した簡単なものが一つだけのシステムでしたので、今のシステムから比べると、かなり幼稚なシステムだったと思います。

その後、何度もバージョンアップを重ね、現在はストキャスティクスや MACD、RCI などの各テクニカル指標（現在 11 個のテクニカル指標プラス任意計算、ろうそく足の形の判断などの指標を実装済み）の計算を組み込み、また時間足も 1 分足、5 分足、10 分足など（時間足は全 11 種）選択できるように改善しています。

売買エンジンも、追加されたテクニカル指標を元にクロスや数値以上、数値以下の場合という判断、時間や過去のデータを使用した判断が出来、それを元にユーザー独自の売買法を構築できるようになりました。

また、完全自動売買ですので、システムを眺めている時間の方が少なくなるのが通常です。システムを稼働させたまま外出もしていますが、注文等が発生した場合には気になります。そのため、注文が発生した場合には携帯にメールが送られてきたり、データの提供元との接続が切断された場合には、それを自動で判断し、自動で再接続を行ったりするようにも改善を加えました。

まだまだ、付加させたい機能もあり、今後もよりよいシステムになって行くと思います。

さて、前回のレポートからこのレポートにかけて、自動売買システムを構築する上での知識やツール、実際にどのような処理を行っているかなどコードを交えて解説してきましたが、中には『そんなまどろっこしいことはせずに、出来合いの自動売買システムを買えばいいじゃないか！』と思われる方もおられると思います。

しかし、いざ自動売買システムを買おうとなると、通常数 10 万円、私が見た最高額のものには 300 万円というのもありました。おそらく安いものでも 5 万円～というのが普通ではないでしょうか。

そのほかにも利用に月額数千円～というものがありますが、長期にわたり使用するとすると、その月額利用料もばかになりません。

確かに、価格と性能は基本的に比例するものだと思います。カスタマイズの可否や扱える相場、操作性などはやはり高いお金を出せばそれなりのものが手に入ります。

しかし、そのような値の張る自動売買システムを買う余力のある方は、そもそもこのレポートは手にとっておられないのではないかと思います。

そこで、現在私が構築した自動売買システムを、自動売買システムの相場では考えられないほどの安価でお譲りしています。

まだまだ発展途上のシステムですので、いろいろと使用の条件に縛りがありますが、十分使えるものだと思いますし、今後更なる進化を遂げる予定です。

ちなみに、『自動売買システムの相場では考えられないほどの安価』ですので、とりあえず1万円を切る値段設定で販売を開始しました。

その後、各種機能を付加し、売買エンジンのカスタマイズも可能となり、現在までに数度の値上げを行っています。といっても、まだまだ格安だとは思いますが・・・2万円でおつりが来る程度です(2007/11/14 現在)。

最終的には少しずつ値上げを行っていきませんが、それでも2万前後にしようと思っています(それ以上になる可能性もあります)。当然、月額制などではなく、今後のバージョンアップもすべて無料ですし、何より、購入者の意見や、要望・希望がシステムに反映されていく体制をとっています。

もし、ご興味のおありの方がおられましたら、下記リンクに詳細がございますので、ご覧頂ければ幸いです。

自動売買システム“UG” → <http://www.ibi-square.jp/fxsys/fxsys1.shtml>

どんなシステムかわからないから…と思われるかたに、体験版も用意しています。こちらは完全な自動売買まではできませんが、データの読み書き、売買サインの発生、売買条件の組立て等は、ほとんど製品版と変わらない内容でご利用いただくことが可能です。

自動売買システム UG 体験版

→ <http://www.ibi-square.jp/fxsys/taiken.shtml>

(尚、体験版をダウンロードされる場合は IBI-Square Stocks 発行の無料メルマガに代理登録されます。無料レポートサイトから等レポートをご請求の場合は、同一メールアドレスにて請求されることにより、重複登録がございませ

ん。別々のアドレスにて請求された場合は両方のアドレスが代理登録されることとなりますが、あらかじめご了承ください。

尚、無料レポートサイトからのアドレスと体験版請求のアドレスとの登録にはタイムブランクがございます。無料レポートサイトでのアドレスと体験版請求時のアドレスが同一のものであっても、登録後メルマガを解除された場合は、体験版請求時のアドレスが再度登録されることもあります。その場合はお手数ですが再解除をお願いいたします。)

以上で当無料レポートは終わりです。自力で自動売買システムを構築するためには、人それぞれの知識や能力にもよりますが、まったくゼロからはじめた場合でも、不可能ではありません。

現に、私はまったくのゼロから始めました。多少時間はかかりましたが、実際に自動売買システムを作り、運用し、成績を残しています。

是非、皆さんもあきらめず挑戦されてください。

銘柄予想配信 IBI-Square Stocks

<http://www.ibi-square.jp/>

-----IBI-Square Stocks の e-book、メルマガ。-----

- 無料レポート『罫線談議 Vol. 001』  
ロウソク足、大陽線・大陰線についての無料レポートです。  
<http://www.ibi-square.jp/keisendangi/001.shtml>
  
- 無料レポート『罫線談議 Vol. 002』  
値上がり率ランキング TOP200 に入った銘柄のその後の株価推移について。  
<http://www.ibi-square.jp/keisendangi/002.shtml>
  
- 無料レポート『罫線談議 Vol. 003』  
移動平均線のゴールデンクロス・デッドクロスについての売買検証。  
<http://www.ibi-square.jp/keisendangi/003.shtml>
  
- 無料レポート『罫線談議 Vol. 004』  
前日比と値下げ率を利用した投資手法とその売買検証。  
<http://www.ibi-square.jp/keisendangi/004.shtml>
  
- 無料レポート『罫線談議 Vol. 005』  
一目均衡表前日比の利用と、売買検証結果。  
<http://www.ibi-square.jp/keisendangi/005.shtml>
  
- メルマガ「あっ、この銘柄あがりそう！！」  
ほぼ日刊の配信を基本に売買シグナルを配信しています。無料版は有料版からの一部抜粋として、有料版ではその日発生したシグナルをすべて配信しています。その他、当日の値上がり率ランキングの掲載、日経平均についての雑感など・・・。  
<http://www.ibi-square.jp/melmaga0.shtml>