



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7

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7弾です。

IBI-Square Stocks イビグチ 著

～IBI-Square Stocks～ 代表：イビグチ ヤスヒロ

URL : <http://www.ibi-square.jp/> Mail : 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 2008 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 -----

こんにちは、IBI-Square Stocks のイビグチといいます。このたびは拙著「自動売買システムを作ってみよう！その7」ダウンロードありがとうございます。

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。

■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。誤解なさらないでほしいのは、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けば、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。

■ 前回までのおさらい

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）とそれらの処理を、その2では実際にデータを読み込み、その3では移動平均線の描画を、その4で移動平均線のクロスでの売買判断、その5では実際に注文を行わせる UWSC のファイルを作成しました。

その4までで作成したエクセルファイルとその5で作成した注文用のキーボードマクロの接続をその6で行い、データの読み込み、書込み、計算、判断、注文の一連の動作を完成させています。

未読の方は↓

「自動売買システムを作ってみよう！」

<http://mailzou.com/get.php?R=4602&M=1142>

「自動売買システムを作ってみよう！その2」

<http://mailzou.com/get.php?R=16188&M=1142>

「自動売買システムを作ってみよう！その3」

<http://mailzou.com/get.php?R=18246&M=1142>

「自動売買システムを作ってみよう！その4」

<http://mailzou.com/get.php?R=19495&M=1142>

「自動売買システムを作ってみよう！その5」

<http://mailzou.com/get.php?R=23873&M=1142>

「自動売買システムを作ってみよう！その6」

<http://mailzou.com/get.php?R=23990&M=1142>

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。

その 2 では上記の処理のうち、①と②について解説をし、具体的なコードを交えて実際に 5 分足のデータを読み込み、4 本値として書き出すプログラムをエクセルを用いて作成する手順を掲載しました。

そして、その 3 にて③の処理、テクニカル指標の計算・書き込みについてが完了しています。その 4 で上記④の売買判断部分が完成しており、その 5 では⑤の「売買の判断によって売買サインが出た場合は注文を実行する」の準備として、注文を行うためのファイル作成を行なっています。

その 6 で、その 5 で作成した注文用のキーボードマクロファイルとこれまでに作成しているデータ読み込み、書き込み、判断ファイルとの接続を行い、上記⑤までの 5 つの処理を完了しています。

今回からは、これまでに作成した処理を更に発展させ、移動平均線クロス以外の売買条件での注文発生を考えていきます。

これまでが「自動売買システムを作ってみよう！（基礎編）」ならば、今回からは「自動売買システムを作ってみよう！（応用編）」となります。

応用編では今後、他の指標や、クロス以外の条件でも売買が出来るように、またチャートの表示や設定値のダイアログ化を行っていきます。

その第 1 弾として、前回の予告とおり、今回はこれまで作成してきた自動売買システムのまとめを行います。

■ ここまでのまとめ

その6まででは下の画像のようなファイルを作成し、時刻の設定を行ったあと、ボタンをクリックすると自動で5分足の4本値を書き込み、移動平均線の計算、クロスでの売買サインの書き込み、保有状態の書き込み、発注を行なうファイルを作成しています。

	A	B	C	D	E	F	G	H	I	J	K
1		売値	買値		行数	移行タイミング					
2		108.07	108.09		60	0					
3											
4											
5	#	始値	高値	安値	終値	日時	移動平均1	移動平均2	買サイン	売サイン	保有
24	19	107.9	107.91	107.89	107.89	2008/07/30 12:39:45	107.906				0
25	20	107.89	107.91	107.88	107.88	2008/07/30 12:44:45	107.896				0
26	21	107.88	107.88	107.87	107.87	2008/07/30 12:49:45	107.89				0
27	22	107.87	107.91	107.87	107.9	2008/07/30 12:54:45	107.89				0
28	23	107.9	107.91	107.9	107.91	2008/07/30 12:59:45	107.89				0
29	24	107.9	107.95	107.9	107.95	2008/07/30 13:04:45	107.902				0
30	25	107.94	107.94	107.9	107.93	2008/07/30 13:09:45	107.912				0
31	26	107.93	107.93	107.92	107.92	2008/07/30 13:14:45	107.922				0
32	27	107.92	107.92	107.91	107.91	2008/07/30 13:19:45	107.924				0
33	28	107.91	107.94	107.91	107.94	2008/07/30 13:24:45	107.93	107.9332			0
34	29	107.93	107.94	107.93	107.93	2008/07/30 13:29:45	107.926	107.9304			0
35	30	107.93	107.94	107.92	107.93	2008/07/30 13:34:45	107.926	107.9276			0
36	31	107.93	107.94	107.93	107.94	2008/07/30 13:39:45	107.93	107.9264	107.94		1
37	32	107.94	107.94	107.93	107.93	2008/07/30 13:44:45	107.934	107.9248			1
38	33	107.93	107.96	107.92	107.96	2008/07/30 13:49:45	107.938	107.9244			1
39	34	107.96	107.97	107.96	107.97	2008/07/30 13:54:45	107.946	107.9248			1
40	35	107.97	107.97	107.96	107.97	2008/07/30 13:59:45	107.954	107.9256			1
41	36	107.96	107.99	107.96	107.99	2008/07/30 14:04:45	107.964	107.928			1
42	37	108	108.01	107.99	107.99	2008/07/30 14:09:45	107.976	107.9296			1
43	38	107.99	107.99	107.97	107.99	2008/07/30 14:14:45	107.982	107.9312			1
44	39	107.98	107.99	107.98	107.99	2008/07/30 14:19:45	107.986	107.9332			1
45	40	107.99	108.01	107.99	108.01	2008/07/30 14:24:45	107.994	107.9364			1
46	41	108.01	108.01	107.99	107.99	2008/07/30 14:29:45	107.994	107.94			1
47	42	107.99	108.02	107.99	108.01	2008/07/30 14:34:45	107.998	107.9444			1
48	43	108.01	108.02	108	108.02	2008/07/30 14:39:45	108.004	107.9488			1

ここまで行なった処理は、まずデータの読み込みを行なっています。

読み込み開始時間と終了時間を設定し、5分足でデータの書き込みを上図のように行なっていく処理を行なったのが「その2」の内容です。

データの読み込みはFXオンラインのデモ口座を用意し、USD/JPYのデータをエクセルに読み込むところから始めています。

読み込みは、セルに直接式を入力し、データを読み込みました。

=myFXDMO|rates!BidUSDJPY

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

上記式と、上図のようなところから開始しています。

その後、自動売買システムの骨格として、データ入力部分の表題を入力しました。

	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							

このときに、行数の表示や移行タイミングといった入力欄も同時に作成しています。

そして、「その2」の一番大きな部分とし、条件分岐を使用しての書込み部分、時間足の判断、更新開始時刻・終了時刻などの処理を行っています。

解説の順番としては、書込み→時間足による条件分岐→更新開始時間・終了時間となっています。

処理は VBA を使用して、4 本値の書きこみを行なう「Sub Kakikomi()」、5 分足の判断を行い、行を移行させるタイミングを計る「Sub Ashi()」を作成しました。

```
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
  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
End Sub
```

```
Sub Ashi()
Calculate
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
Call IdouHeikin
Call Handan

End Sub
```

次に、タスクを使用して、更新開始時刻と終了時刻、更新間隔などを設定し、それに則って更新を行なう部分、「Sub TascBigen()」の作成を行なっています。

```

Dim mcolTask As Collection

Sub Tasc_Bigin()
' セッティング-----
  dayBigin = "2008/10/07"           ' 開始日付
  datBigin = TimeValue("08:00:00") ' 開始時刻
  dayEnd = "2008/10/07"            ' 終了日付
  datEnd = TimeValue("22:00:00")   ' 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

```

そして、稼動を開始するためのボタンを配置したところまでが「その2」の内容となります。

	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								CommandButton1	
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								

次の「その3」では、移動平均線の計算と書込みを行なっています。
まず、移動平均線を入力する場所に表題をつけました。

	A	B	C	D	E	F	G	H
1		売値	買値		行数	移行タイミング		
2		110.93	110.96		28	1		
3								CommandButton1
4								
5	#	始値	高値	安値	終値	日時	移動平均1	移動平均2
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 10:29:45		
15	10	111.11	111.12	111.03	111.03	2007/11/14 10:34:45		
16	11	111.02	111.05	111.01	111.05	2007/11/14 10:39:45		
17	12	111.05	111.07	111.04	111.05	2007/11/14 10:44:45		
18	13	111.06	111.06	111.04	111.05	2007/11/14 10:49:45		
19	14	111.05	111.07	111.04	111.04	2007/11/14 10:54:45		
20	15	111.05	111.06	111.02	111.04	2007/11/14 10:59:45		
21	16	111.03	111.04	110.98	110.98	2007/11/14 11:04:45		
22	17	110.97	110.99	110.94	110.96	2007/11/14 11:09:45		
23	18	110.96	110.96	110.88	110.91	2007/11/14 11:14:45		
24	19	110.91	110.93	110.89	110.89	2007/11/14 11:19:45		
25	20	110.88	110.89	110.84	110.89	2007/11/14 11:24:45		
26	21	110.91	110.93	110.85	110.85	2007/11/14 11:29:45		
27	22	110.85	110.94	110.85	110.93	2007/11/14 11:34:45		
28	23	110.93	110.95	110.91	110.93	2007/11/14 11:38:15		

VBA で移動平均線の計算を行う「Sub IdouHeikin()」の作成も「その3」で行なっています。

```

Sub IdoHeikin()
ma1 = 5
ma2 = 25

gyou = Cells(2, 5)

If gyou >= ma1 + 8 Then
Cells(gyou, 7) = Application.WorksheetFunction.Average(Range(Cells(gyou - (ma1 - 1), 5), Cells(gyou, 5)))
End If

If gyou >= ma2 + 8 Then
Cells(gyou, 8) = Application.WorksheetFunction.Average(Range(Cells(gyou - (ma2 - 1), 5), Cells(gyou, 5)))
End If

End Sub
    
```

「その4」になり、移動平均線のクロスでの売買サインの発生を行っています。

買サイン、売サインの欄を作成したのも「その4」でのことです

#	始値	高値	安値	終値	日時	移動平均1	移動平均2	買サイン	売サイン	保有
1	101.55	101.59	101.4	101.59	2008/10/07 09:19:45					0
2	101.58	101.75	101.57	101.75	2008/10/07 09:24:45					0
3	101.74	101.78	101.62	101.62	2008/10/07 09:29:45					0
4	101.9	101.81	101.6	101.66	2008/10/07 09:34:45					0
5	102.06	101.84	101.66	101.79	2008/10/07 09:39:45					0
6	102.22	101.87	101.69	101.73	2008/10/07 09:44:45					0
7	102.38	101.9	101.66	101.69	2008/10/07 09:49:45					0
8	102.54	101.93	101.63	101.64	2008/10/07 09:54:45	101.702				0
9	102.7	101.96	101.52	101.54	2008/10/07 09:59:45	101.678				0
10	102.86	101.99	101.29	101.33	2008/10/07 10:04:45	101.586				0
11	101.32	101.46	101.32	101.43	2008/10/07 10:09:45	101.526				0
12	101.43	101.58	101.43	101.48	2008/10/07 10:14:45	101.484				0
13	101.46	101.55	101.43	101.53	2008/10/07 10:19:45	101.462				0
14	101.53	101.6	101.53	101.57	2008/10/07 10:24:45	101.468				0
15	101.54	101.56	101.49	101.54	2008/10/07 10:29:45	101.51				0
16	101.53	101.62	101.5	101.62	2008/10/07 10:34:45	101.548				0

そして、移動平均線のゴールデンクロス、デッドクロスでのドテン売買を行うべく、「Sub Handan()」というコードを作成しました。

```

Sub Handan()

gyou = Cells(2, 5)
ma05_b = Cells(gyou - 1, 7)
ma05_n = Cells(gyou, 7)
ma25_b = Cells(gyou - 1, 8)
ma25_n = Cells(gyou, 8)

r_now = Val(Cells(2, 2))

If Cells(gyou, 9) = "" And Cells(gyou, 10) = "" Then
    Cells(gyou, 11) = Cells(gyou - 1, 11)
End If

If Cells(gyou - 1, 11) = "保有" Then Cells(gyou, 11) = 0
If Cells(gyou - 1, 11) = "" Then Cells(gyou, 11) = 0

If Cells(gyou - 1, 8) <> "" Then

    If Cells(gyou, 9) = "" Then
        If ma05_b < ma25_b And ma05_n > ma25_n Then
            Cells(gyou, 9) = r_now
            Cells(gyou, 11) = 1
            flg_b = 1
        End If
    End If

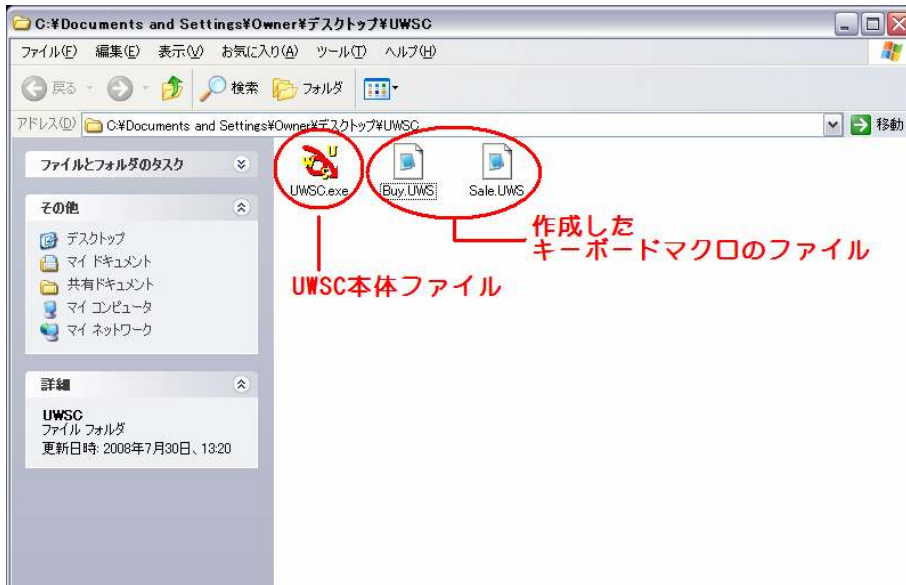
    If Cells(gyou, 10) = "" Then
        If ma05_b > ma25_b And ma05_n < ma25_n Then
            Cells(gyou, 10) = r_now
            Cells(gyou, 11) = -1
            flg_s = 1
        End If
    End If
End If

```

保有状況の書き込みを行なったのもこの部分です。

「その5」では、VBA は全く使用せずに、UWSC のダウンロードと、注文用キーボードマクロの作成として、UWSC での記録やメモ帳での編集などを行なっています。





そして、前回の「その6」にて「その5」で作成したUWSCファイルと「その4」までで作成していたエクセルファイルの接続を行うために、「Sub

Handan()」に買フラッグや売フラッグを追加し、

```

If ma05_b < ma25_b And ma05_n > ma25_n Then
    Cells(zyou, 9) = r_now
    Cells(zyou, 11) = 1
    flg_b = 1
End If

If ma05_b > ma25_b And ma05_n < ma25_n Then
    Cells(zyou, 10) = r_now
    Cells(zyou, 11) = -1
End If
    
```

——— 買いサイン発生でflg_bは1

そのフラッグの条件分岐にて UWSC の実行を行なうための Shell 関数部分を作成しました。

```

If flg_b = 1 Or flg_s = 1 Then

    If flg_b = 1 Then file_name = "Buy.UWS"
    If flg_s = 1 Then file_name = "Sale.UWS"

    pr_name = "C:\UWSC\UWSC.exe "
    path_name = "C:\UWSC\"
    Shell (pr_name & " " & path_name & file_name)

    If Cells(zyou - 1, 11) <> 0 Then
        Shell (pr_name & " " & path_name & file_name)
        Application.Wait Now + TimeValue("00:00:15")
    End If

    xxx = Shell(pr_name & path_name & file_name, vbNormalNoFocus)

End If
End If
End Sub
    
```

これにより、データの読み込み、書込み、移動平均線の計算、売買判断、実際の注文を自動で行うシステムを完成させています。

	A	B	C	D	E	F	G	H	I	J	K	L
1		売値	買値		行數	移行タイミング						
2		108.38	108.4		55	1						
3												
4												
5	#	始値	高値	安値	終値	日時	移動平均1	移動平均2	買サイン	売サイン	保有	
34	29	108.38	108.38	108.37	108.38	2008/08/06 11:44:45	108.376	108.3832			0	
35	30	108.38	108.38	108.35	108.36	2008/08/06 11:49:45	108.374	108.3816			0	
36	31	108.35	108.35	108.34	108.34	2008/08/06 11:54:45	108.366	108.3796			0	
37	32	108.34	108.35	108.33	108.34	2008/08/06 11:59:45	108.36	108.3772			0	
38	33	108.34	108.35	108.33	108.34	2008/08/06 12:04:45	108.352	108.374			0	
39	34	108.34	108.34	108.33	108.33	2008/08/06 12:09:45	108.342	108.37			0	
40	35	108.33	108.33	108.32	108.32	2008/08/06 12:14:45	108.334	108.3648			0	
41	36	108.32	108.33	108.32	108.33	2008/08/06 12:19:45	108.332	108.36			0	
42	37	108.33	108.34	108.31	108.34	2008/08/06 12:24:45	108.332	108.356			0	
43	38	108.33	108.34	108.32	108.32	2008/08/06 12:29:45	108.328	108.352			0	
44	39	108.32	108.33	108.32	108.33	2008/08/06 12:34:45	108.328	108.3484			0	
45	40	108.33	108.34	108.33	108.33	2008/08/06 12:39:45	108.33	108.346			0	
46	41	108.33	108.33	108.31	108.32	2008/08/06 12:44:45	108.328	108.3456			0	
47	42	108.32	108.32	108.29	108.29	2008/08/06 12:49:45	108.318	108.344			0	
48	43	108.29	108.32	108.29	108.31	2008/08/06 12:54:45	108.316	108.3424			0	
49	44	108.31	108.34	108.31	108.33	2008/08/06 12:59:45	108.316	108.3424			0	
50	45	108.33	108.33	108.31	108.31	2008/08/06 13:04:45	108.312	108.3408			0	
51	46	108.3	108.31	108.28	108.31	2008/08/06 13:09:45	108.31	108.3396			0	
52	47	108.32	108.32	108.31	108.31	2008/08/06 13:14:45	108.314	108.3384			0	
53	48	108.31	108.34	108.31	108.34	2008/08/06 13:19:45	108.32	108.338			0	
54	49	108.34	108.37	108.33	108.37	2008/08/06 13:24:45	108.328	108.338			0	
55	50	108.37	108.39	108.37	108.38	2008/08/06 13:27:45	108.342	108.3384	108.37		1	
56												
57												
58												

↑サインの発生

↓注文の実行

The screenshot shows the FX Trading Station 2.0 interface. The main window displays market data for various currency pairs (USD/JPY, EUR/USD, GBP/USD, USD/CHF) with bid/ask prices and spreads. On the right, there are panels for '口座情報' (Account Information) and '注文状況' (Order Status). At the bottom, the 'ポジション情報' (Position Information) panel is visible, showing a table of open positions. A red circle highlights a position for '1-ドル' (1 Dollar) with a quantity of 10, a buy price of 108.381, and a sell price of 108.398.

通貨ペア	口座番号	通貨	取引額	買値	売値	決済レート	ストップ	トレイリングストップ	ロット	損益	差損益	スワップ	時間
1-ドル	00199596	USD/JPY	10 B	108.381	108.398					1.7	170	0	2008/06

さて、今回は「応用編」の入り口として、ここまでのおさらいをやってきましたが、どのくらい覚えておられたでしょうか？

この後は、仕掛けの条件や損切り設定、手仕舞い条件など、色々な組合せや、指標の追加など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。ある程度の知識がたまり、次回以降解説予定の指標の追加や各種売買判断などの実装を経験し、実際に稼働させると、「こういう風な条件判断をさせたい」など良い意味での「欲」が出てきます。

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。

現在は、その後の幾度ものバージョンアップを経て、それらを元にした新システムを完成させました。既に新システムになってからも20回を超えるバージョンアップを行っており、その都度、指標や機能の追加を行なっています。

今回、最後の価格改定を予定（08年10月13日に予定）しており、「自動売買システムUG」としても、今後の更なる発展に向け動き出しています。

現在、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21種30パターン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、時間による売買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12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（因みに、VBAのコードを書くような条件設定はいりません。すべてインターフェイスにより視覚的に条件設定を可能です）、それを元にユーザー独自の売買法の構築は勿論ですが、複数の時間足を売買判断に用いることも可能です。

例えば、30分足でRCIが80%以上の時に、5分足の移動平均線デッドクロスで売るなどの、通常監視下でしか実現し得なかった売買法の設定が可能です。

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。

現在実装されている機能は、

- 完全自動売買(UWSCによる自動発注)
- ユーザー設定による任意の売買条件設定
- 各種時間足設定(全 21 種)
- 再計算(バックテスト)機能
- 損切り監視(全 5 種)
- 各種テクニカル指標(12 種 16 パターン)
- ユーザー設定による任意計算(最大 12)
- ロウソク足の形による売買
- 自動復帰(サーバー切断によるレート取得停止時)
- 休止時間設定(昼休み等のレート更新を停止)
- 最大同時取引数 10 取引
- 対応金商品: 為替、N225、N225mini、株式(東証、大証、JASDAQ、ヘラクレス)

・・・などなど

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も 300 万円というものもありました。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なると、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。現在の最新版では、前頁で一覧にした機能などが入っており、旧バージョンから比べてもインターフェイスや全体的な見易さ、操作のしやすさ、且つ細かな設定が出来るようになりました。

現在は2.5万円での販売となっていますが、先述の通り10月13日に値上げを実行し、その後は¥29,800となる予定です。今回の価格改定を最終と位置づけ、今後はUGの販売は抑え、機能的に更なる上を目指して開発に臨むこととなります。

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。(上記新システムの体験版です)

自動売買システム UG 体験版

→ 上記リンクの中ほどにあります。

(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。

また、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/>

■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。無料版は有料版の結果のみですが、有料版も1,500/月で、配信銘柄の殆どは7営業日以内に数%の利益を得ていますので、一度の取引で元は取れる価格にしています。その他、当日の値上がり率ランキングの掲載など・・・。

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