重播盤勢的「模擬操盤系統」與「離線模擬下單系統」

❑模擬系統操作

由於教學過程中，屬於非開盤期間，即時下單系統範例無法接收即時資料，因此設計一可以「選擇性的播放特定日期」的「模擬操盤系統」與「離線模擬下單系統」，除了選擇性的重播盤勢，系統亦可設定播放速度。

此系統中包含兩大功能：

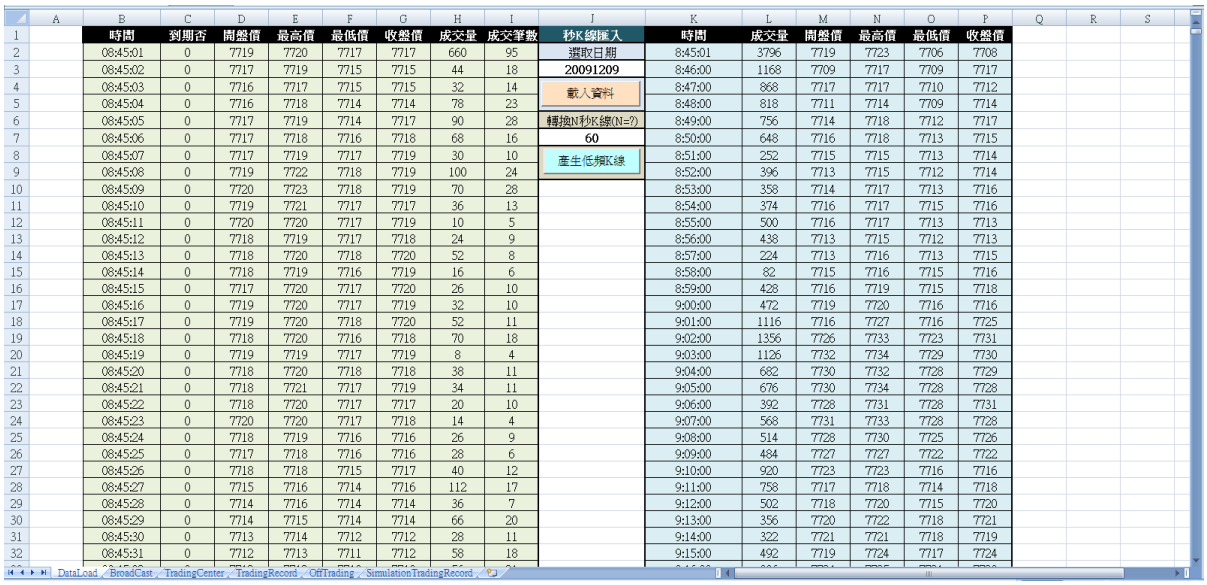
1. 做為**「模擬操盤系統」(或稱盤感訓練系統)**，於播放盤勢時，可以在模擬盤中手動決定買賣，以訓練交易員找出盤中交易規律。
2. 做為**「離線模擬下單系統」**，此系統以讀入後預存放於工作表中的秒 K 線，逐一定頻讀取產生低頻 K 線，並驅動交易策略下單，系統中亦模擬下單 API 的函數呼叫，以期與真實系統做到最近似的無縫連接。

此系統包含 6 個工作表，每一功能表的功能如表 1 所列。

表 1 離線模擬下單系統工作表說明

工作表	內容
DataLoad	此工作表如圖 1。可於 J3 格位中設定特定日期，按下「載入資料」按鍵，即可將當日的「秒 K 線」資料匯入(每天共 18,000 筆)，「秒 K 線」資料匯入後存於「B2:I18001」範圍格位中；匯入資料分別為時間(B 欄)、到期日否標籤(0 表非到期日，1 表到期日) (C 欄)、開盤價(D 欄)、最高價(E 欄)、最低價(F 欄)、收盤價(G 欄)、成交量(H 欄)、成交筆數(I 欄)。 資料匯入後，可於 J7 格位中設定轉為其他秒數 K 線資料(例如設定 60 秒，即可得分 K 線)，設定後，按下「產生低頻 K 線」按鍵，即可將秒 K 線轉為其他低頻 K 線並存於 K 欄到 P 欄中，分別為時間(K 欄)、成交量(L 欄)、開盤價(M 欄)、最高價(N 欄)、最低價(O 欄)、收盤價(P 欄)。 秒 K 線資料將提供「離線模擬下單系統」之用，低頻 K 線資料將提供「離線模擬下單系統」之用。
「離線模擬下單系統」	
Broadcast	此工作表如圖 2。此工作表由「TradingCenter」工作表之「開始交易」按鍵驅動，定時(由 SetTimer 函數驅動)將 K 線資料搬移到「Broadcast」工作表中，如此即可模擬 K 線產生的過程。
TradingCenter	1. 此工作表如圖 3。在此工作表中可以「開始交易」按鍵驅動交易，以「結束交易」按鍵結束交易，在交易期間可以「市價買進」按鍵作多 1 口期貨，以「市價賣出」按鍵作空 1 口期貨。 2. 開始交易前，可於 B3 格位設定 K 線播放速度，B4 格位計算 300 根 K 線的模擬時間(格位公式為「=300/B3」，單位為秒)，B5 格位計算模擬期間實際交易時間(格位公式為「=300*B2/60」，單位為分)，D1 格位設定買進部位上限，D2 格位設定賣出部位上限，D3 格位設定手續費，D4 格位設定交易稅。 3. 交易期間，L1 格位可以隨時顯示目前的交易部位，N1 格位可以隨時顯示目前最新期貨價位，L2 格位可以隨時顯示目前部位的帳戶價值，N2 格位以格位公式計算部位價值(公式為「=200*L1*N1」)，N3 格位以格位公式計算累積損益(公式為「=L2+N2」)。
TradingRecord	此工作表如圖 4。可於「TradingCenter」工作表交易期間，紀錄交易時間、買賣口數、交易量、累積部位、累積損益等資訊，並以程式碼計算交易次數(I5 格位)、買進次數(I6 格位)、賣出次數(I7 格位)、最後損益(I8 格位)、最高獲益(I9 格位)、最低獲利(I10 格位)，以及繪出累積損益圖。
「離線模擬下單系統」	
OffTrading	1.此工作表如圖 5。可以「開始模擬自動交易」按鍵驅動離線模擬下單交易，以「終止模擬交易」按鍵中斷模擬下單交易。R2 格位顯示為目前的交易部位，X2 格位設定短(快速)均線參數，

	X3 格位設定長(慢速)均線參數，Z2 格位設定 RSI 回算天數，Z3 格位設定 RSI 買進下限值，Z4 格位設定 RSI 賣出上限值，AB1 格位以格位公式計算回算保留天數，格位公式為「=MAX(X3+1,Z2)」(即「長均線 +1」與「RSI 回算天數」兩者取其大)。 2.工作表於模擬下單交易期間，同步顯示「K 線圖」與「損益圖」。
SimulationTradingRecord	此工作表如圖 6。當 OffTrading 工作表模擬期間同步輸出下單回報資訊於此工作表中，顯示資訊包括交易商品、買賣方向、買賣單位、買賣價格、淨損益。

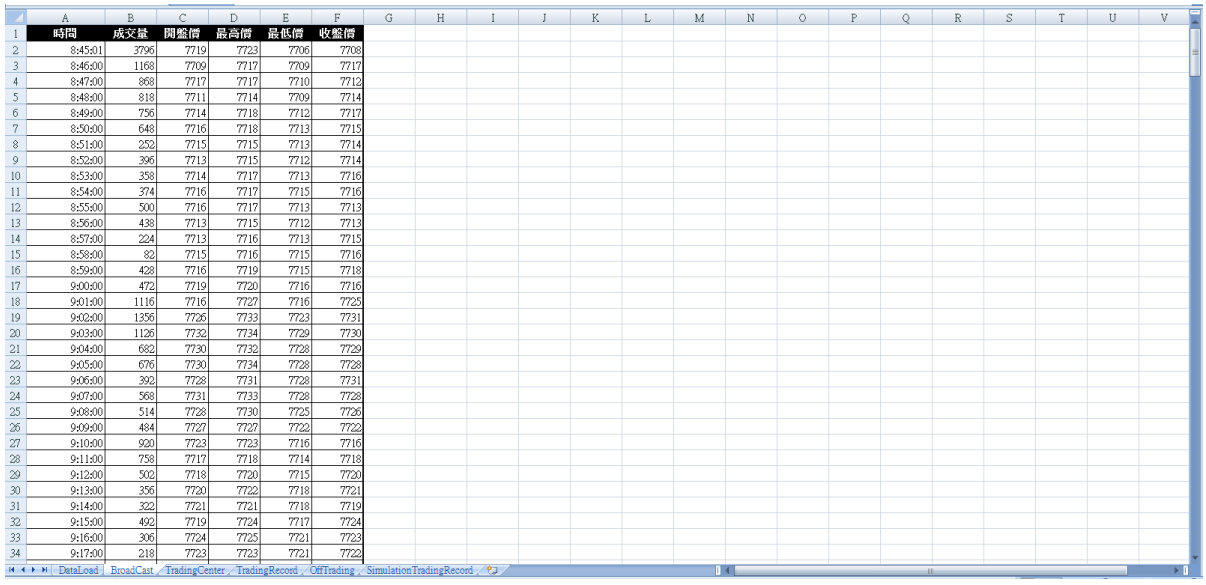


時間	到期否	開盤價	最高價	最低價	收盤價	成交量	成交筆數
08:45:01	0	7719	7720	7717	7717	660	95
08:45:02	0	7717	7719	7715	7715	44	18
08:45:03	0	7716	7717	7715	7715	32	14
08:45:04	0	7716	7718	7714	7714	78	23
08:45:05	0	7717	7719	7714	7717	90	28
08:45:06	0	7717	7718	7716	7718	68	16
08:45:07	0	7717	7719	7717	7719	30	10
08:45:08	0	7719	7722	7718	7719	100	24
08:45:09	0	7720	7723	7718	7719	70	28
08:45:10	0	7719	7721	7717	7717	36	13
08:45:11	0	7720	7720	7717	7719	10	5
08:45:12	0	7718	7719	7717	7718	24	9
08:45:13	0	7718	7720	7718	7720	52	8
08:45:14	0	7718	7719	7716	7719	16	6
08:45:15	0	7717	7720	7717	7720	26	10
08:45:16	0	7719	7720	7717	7719	32	10
08:45:17	0	7719	7720	7718	7720	52	11
08:45:18	0	7718	7720	7716	7718	70	18
08:45:19	0	7719	7719	7717	7719	8	4
08:45:20	0	7718	7720	7718	7718	38	11
08:45:21	0	7718	7721	7717	7719	34	11
08:45:22	0	7718	7720	7717	7717	20	10
08:45:23	0	7720	7720	7717	7718	14	4
08:45:24	0	7718	7719	7716	7716	26	9
08:45:25	0	7717	7718	7716	7716	28	6
08:45:26	0	7718	7718	7715	7717	40	12
08:45:27	0	7715	7716	7714	7716	112	17
08:45:28	0	7714	7716	7714	7714	36	7
08:45:29	0	7714	7715	7714	7714	66	20
08:45:30	0	7713	7714	7712	7712	28	11
08:45:31	0	7712	7713	7711	7712	58	18

秒K線匯入
選取日期
20091209
載入資料
轉換N秒K線(N=?)
60
產生低頻K線

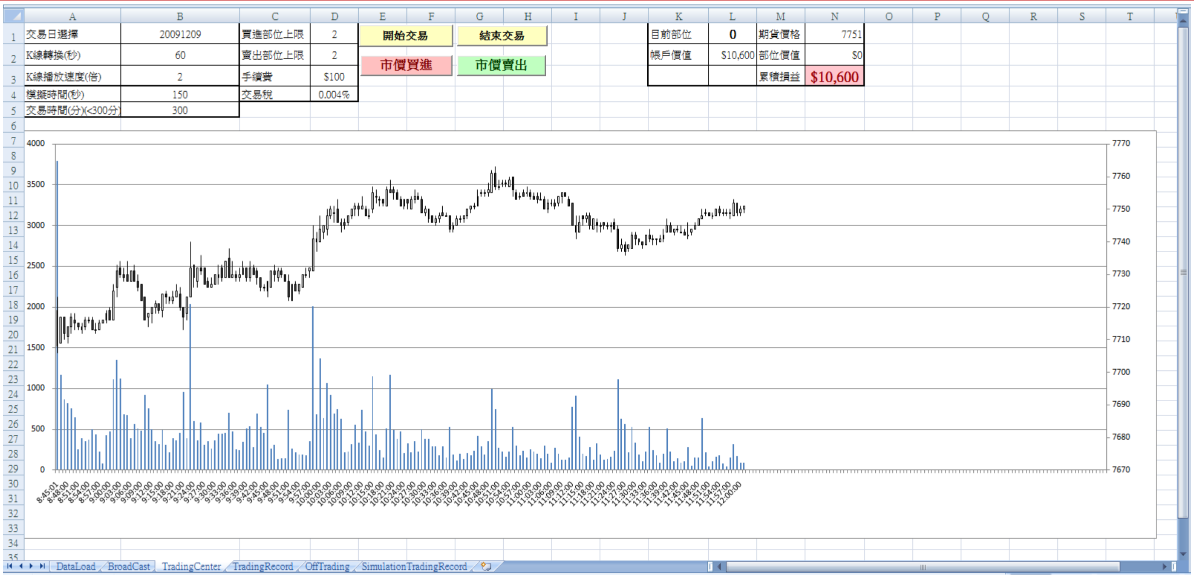
時間	成交量	開盤價	最高價	最低價	收盤價
8:45:01	3796	7719	7723	7706	7708
8:46:00	1168	7709	7717	7709	7717
8:47:00	868	7717	7717	7710	7712
8:48:00	818	7711	7714	7709	7714
8:49:00	756	7714	7718	7712	7717
8:50:00	648	7716	7718	7713	7715
8:51:00	252	7715	7715	7713	7714
8:52:00	396	7713	7715	7712	7714
8:53:00	358	7714	7717	7713	7716
8:54:00	374	7716	7717	7715	7716
8:55:00	500	7716	7717	7713	7713
8:56:00	438	7713	7715	7712	7713
8:57:00	224	7713	7716	7713	7715
8:58:00	82	7715	7716	7715	7716
8:59:00	428	7716	7719	7715	7718
9:00:00	472	7719	7720	7716	7716
9:01:00	1116	7716	7727	7716	7725
9:02:00	1356	7726	7733	7723	7731
9:03:00	1126	7732	7734	7729	7730
9:04:00	682	7730	7732	7728	7729
9:05:00	676	7730	7734	7728	7728
9:06:00	392	7728	7731	7728	7731
9:07:00	568	7731	7733	7728	7728
9:08:00	514	7728	7730	7725	7726
9:09:00	484	7727	7727	7722	7722
9:10:00	920	7723	7723	7716	7716
9:11:00	758	7717	7718	7714	7718
9:12:00	502	7718	7720	7715	7720
9:13:00	356	7720	7722	7718	7721
9:14:00	322	7721	7721	7718	7719
9:15:00	492	7719	7724	7717	7724

圖 1 「DataLoad」工作表



時間	成交量	開盤價	最高價	最低價	收盤價
8:45:01	3796	7719	7723	7706	7708
8:46:00	1168	7709	7717	7709	7717
8:47:00	868	7717	7717	7710	7712
8:48:00	818	7711	7714	7709	7714
8:49:00	756	7714	7718	7712	7717
8:50:00	648	7716	7718	7713	7715
8:51:00	252	7715	7715	7713	7714
8:52:00	396	7713	7715	7712	7714
8:53:00	358	7714	7717	7713	7716
8:54:00	374	7716	7717	7715	7716
8:55:00	500	7716	7717	7713	7713
8:56:00	438	7713	7715	7712	7713
8:57:00	224	7713	7716	7713	7715
8:58:00	82	7715	7716	7715	7716
8:59:00	428	7716	7719	7715	7718
9:00:00	472	7719	7720	7716	7716
9:01:00	1116	7716	7727	7716	7725
9:02:00	1356	7726	7733	7723	7731
9:03:00	1126	7732	7734	7729	7730
9:04:00	682	7730	7732	7728	7729
9:05:00	676	7730	7734	7728	7728
9:06:00	392	7728	7731	7728	7731
9:07:00	568	7731	7733	7728	7728
9:08:00	514	7728	7730	7725	7726
9:09:00	484	7727	7727	7722	7722
9:10:00	920	7723	7723	7716	7716
9:11:00	758	7717	7718	7714	7718
9:12:00	502	7718	7720	7715	7720
9:13:00	356	7720	7722	7718	7721
9:14:00	322	7721	7721	7718	7719
9:15:00	492	7719	7724	7717	7724
9:16:00	306	7724	7725	7721	7723
9:17:00	218	7723	7723	7721	7722

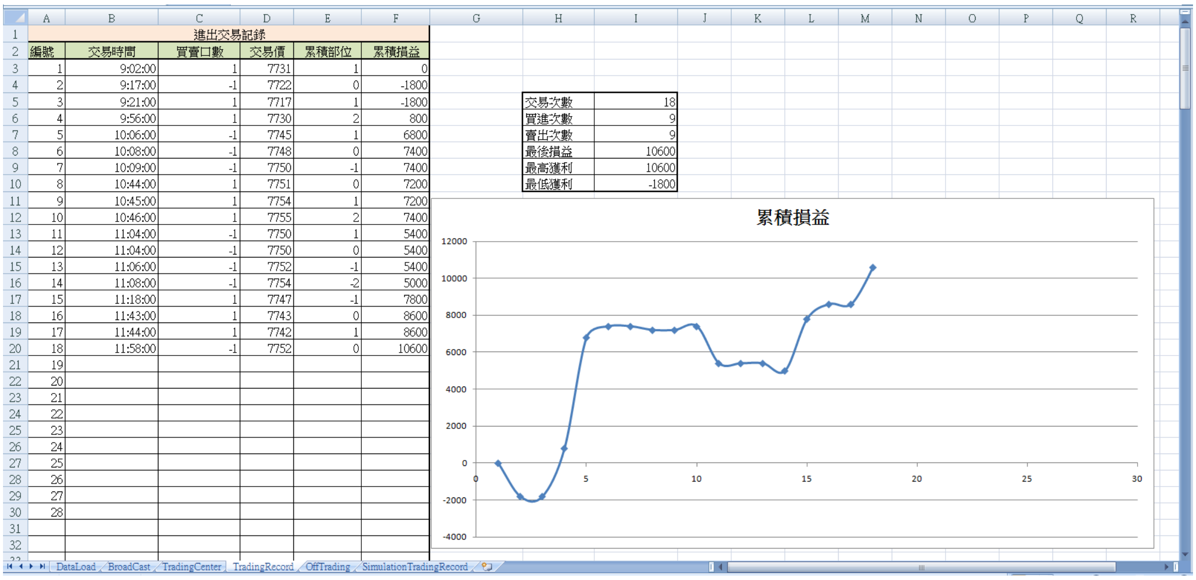
圖 2「Broadcast」工作表



交易日選擇	20091209	買進部位上限	2
K線轉換(秒)	60	賣出部位上限	2
K線播放速度(倍)	2	手續費	$100
模擬時間(秒)	150	交易稅	0.004%
交易時間(分)(<300分)	300		

目前部位	0	期貨價格	7751
帳戶價值	$10,600	部位價值	$0
		累積損益	$10,600

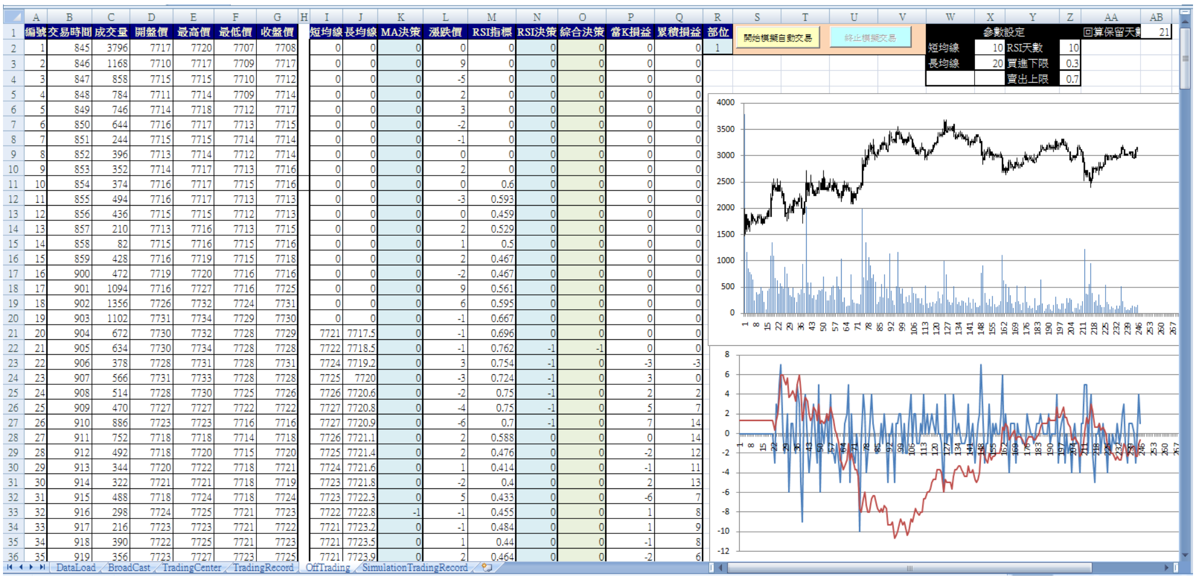
圖 3「TradingCenter」工作表



進出交易記錄					
編號	交易時間	買賣口數	交易價	累積部位	累積損益
1	9:02:00	1	7731	1	0
2	9:17:00	-1	7722	0	-1800
3	9:21:00	1	7717	1	-1800
4	9:56:00	1	7730	2	800
5	10:06:00	-1	7745	1	6800
6	10:08:00	-1	7748	0	7400
7	10:09:00	-1	7750	-1	7400
8	10:44:00	1	7751	0	7200
9	10:45:00	1	7754	1	7200
10	10:46:00	1	7755	2	7400
11	11:04:00	-1	7750	1	5400
12	11:04:00	-1	7750	0	5400
13	11:06:00	-1	7752	-1	5400
14	11:08:00	-1	7754	-2	5000
15	11:18:00	1	7747	-1	7800
16	11:43:00	1	7743	0	8600
17	11:44:00	1	7742	1	8600
18	11:58:00	-1	7752	0	10600
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					

交易次數	18
買進次數	9
賣出次數	9
最後損益	10600
最高獲利	10600
最低獲利	-1800

圖 4「TradingRecord」工作表

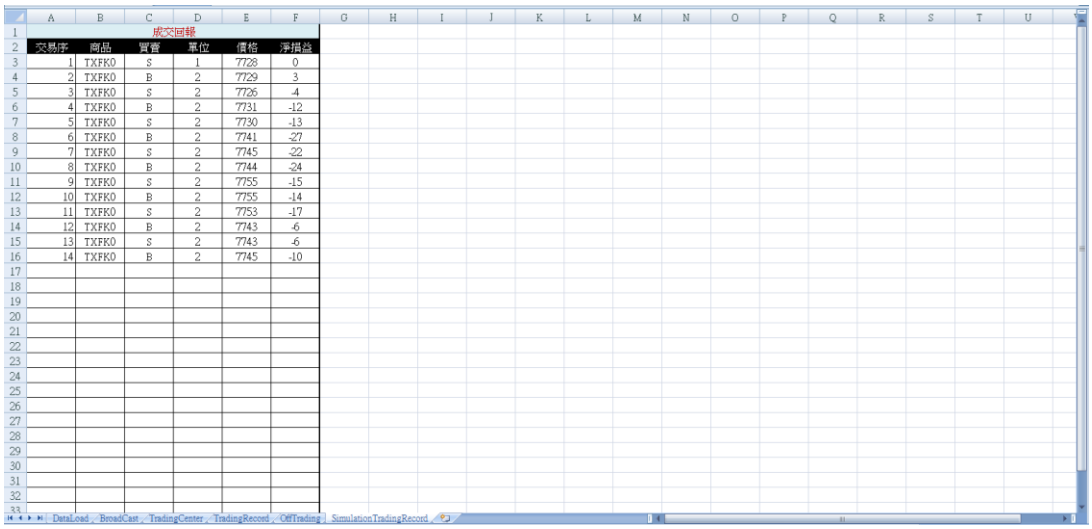


編號	交易時間	成交量	開盤價	最高價	最低價	收盤價	短均線	長均線	MA決策	漲跌價	RSI指標	RSI決策	綜合決策	當K損益	累積損益	部位
1	845	3796	7717	7720	7707	7708	0	0	0	0	0	0	0	0	0	1
2	846	1168	7710	7717	7709	7717	0	0	0	9	0	0	0	0	0	
3	847	858	7715	7715	7710	7712	0	0	0	-5	0	0	0	0	0	
4	848	784	7711	7714	7709	7714	0	0	0	2	0	0	0	0	0	
5	849	746	7714	7718	7712	7717	0	0	0	3	0	0	0	0	0	
6	850	644	7716	7717	7713	7715	0	0	0	-2	0	0	0	0	0	
7	851	244	7715	7715	7714	7714	0	0	0	-1	0	0	0	0	0	
8	852	396	7713	7714	7712	7714	0	0	0	0	0	0	0	0	0	
9	853	352	7714	7717	7713	7716	0	0	0	2	0	0	0	0	0	
10	854	374	7716	7717	7715	7716	0	0	0	0	0.6	0	0	0	0	
11	855	494	7716	7717	7713	7713	0	0	0	-3	0.593	0	0	0	0	
12	856	436	7715	7715	7712	7713	0	0	0	0	0.459	0	0	0	0	
13	857	210	7713	7716	7713	7715	0	0	0	2	0.529	0	0	0	0	
14	858	82	7715	7716	7715	7716	0	0	0	1	0.5	0	0	0	0	
15	859	428	7716	7719	7715	7718	0	0	0	2	0.467	0	0	0	0	
16	900	472	7719	7720	7716	7716	0	0	0	-2	0.467	0	0	0	0	
17	901	1094	7716	7727	7716	7725	0	0	0	9	0.561	0	0	0	0	
18	902	1356	7726	7732	7724	7731	0	0	0	6	0.595	0	0	0	0	
19	903	1102	7731	7734	7729	7730	0	0	0	-1	0.667	0	0	0	0	
20	904	672	7730	7732	7728	7729	7721	7717.5	0	-1	0.696	0	0	0	0	
21	905	634	7730	7734	7728	7728	7722	7718.5	0	-1	0.762	-1	-1	0	0	
22	906	378	7728	7731	7728	7731	7724	7719.2	0	3	0.754	-1	0	-3	-3	
23	907	566	7731	7733	7728	7728	7725	7720	0	-3	0.724	-1	0	3	0	
24	908	514	7728	7730	7725	7726	7726	7720.6	0	-2	0.75	-1	0	2	2	
25	909	470	7727	7727	7722	7722	7727	7720.8	0	-4	0.75	-1	0	5	7	
26	910	886	7723	7723	7716	7716	7727	7720.9	0	-6	0.7	-1	0	7	14	
27	911	752	7718	7718	7714	7718	7726	7721.1	0	2	0.588	0	0	0	14	
28	912	492	7718	7720	7715	7720	7725	7721.4	0	2	0.476	0	0	-2	12	
29	913	344	7720	7722	7718	7721	7724	7721.6	0	1	0.414	0	0	-1	11	
30	914	322	7721	7721	7718	7719	7723	7721.8	0	-2	0.4	0	0	2	13	
31	915	488	7718	7724	7718	7724	7723	7722.3	0	5	0.433	0	0	-6	7	
32	916	298	7724	7725	7721	7723	7722	7722.8	-1	-1	0.455	0	0	1	8	
33	917	216	7723	7723	7721	7722	7721	7723.2	0	-1	0.484	0	0	1	9	
34	918	390	7722	7725	7721	7723	7721	7723.5	0	1	0.44	0	0	-1	8	
35	919	356	7723	7727	7723	7725	7721	7723.9	0	2	0.464	0	0	-2	6	

參數設定			
短均線	10	RSI天數	10
長均線	20	買進下限	0.3
		賣出上限	0.7

回算保留天數	21

圖 5「OffTrading」工作表



成交回報					
交易序	商品	買賣	單位	價格	淨損益
1	TXFK0	S	1	7728	0
2	TXFK0	B	2	7729	3
3	TXFK0	S	2	7726	-4
4	TXFK0	B	2	7731	-12
5	TXFK0	S	2	7730	-13
6	TXFK0	B	2	7741	-27
7	TXFK0	S	2	7745	-22
8	TXFK0	B	2	7744	-24
9	TXFK0	S	2	7755	-15
10	TXFK0	B	2	7755	-14
11	TXFK0	S	2	7753	-17
12	TXFK0	B	2	7743	-6
13	TXFK0	S	2	7743	-6
14	TXFK0	B	2	7745	-10

圖 6「SimulationTradingRecord」工作表

「離線模擬下單系統」程式碼解析

當 OffTrading 工作表之「開始模擬自動交易」按鍵 Click 時，設定按鍵屬性，並呼叫「Start_ DDE_Auto_Order」程序

```
1  Private Sub BeginTrade_Click() '「開始模擬自動交易」按鍵之 Click 事件
2  StopTrade.Enabled = True   '啟動「終止模擬交易」按鍵 Enabled 屬性
3  Sheets("OffTrading").Range("B2:Q301").ClearContents   '清除 OffTrading 工作表之資料區
4  Call Start_DDE_Auto_Order   '呼叫「Start_ DDE_Auto_Order」程序
5  BeginTrade.Enabled = False   '關閉「開始模擬自動交易」按鍵 Enabled 屬性
6  End Sub
```

當 OffTrading 工作表之「終止模擬交易」按鍵 Click 時，設定按鍵屬性，並呼叫「Close_ DDE_Auto_Order」程序

```
1  Private Sub StopTrade_Click() '「終止模擬交易」按鍵之 Click 事件
2  StopTrade.Enabled = False   '關閉「終止模擬交易」按鍵 Enabled 屬性
3  Call Close_DDE_Auto_Order   '呼叫「Close_ DDE_Auto_Order」程序
4  BeginTrade.Enabled = True   '啟動「開始模擬自動交易」按鍵 Enabled 屬性
5  End Sub
```

VBA 模組(Module2)之「一般宣告區」宣告

```
1  Private Declare Function SetTimer Lib "user32" (ByVal hWnd As Long, ByVal nIDEvent As Long, ByVal uElapse As Long, ByVal
   lpTimerFunc As Long) As Long   '宣告 SetTimer 函數以使用 Windows API 函式的啟動計時器事件
2  Private Declare Function KillTimer Lib "user32" (ByVal hWnd As Long, ByVal nIDEvent As Long) As Long
   '宣告 KillTimer 函數以使用 Windows API 函式的關閉計時器事件
3  Dim hTimer As Variant   '宣告計時器傳回值
4  Dim BarNo As Integer, SheetsRow As Integer, Initial_Flag As Integer, DataRow As Long, TradeNo As Long, BeginTradeNo As Integer
5  Dim FMA_P As Integer, SMA_P As Integer, RSI_P As Integer, RSI_B As Single, RSI_S As Single
6  Dim Min_Bar(300, 17) As Single   '此陣列用以儲存模擬交易過程的資料，陣列資料將輸出到「A301:Q301」範圍格位
7  Dim PreTickTime As Single   '儲存前一 TickData 的時間
8  Dim Position As Integer   ' Position 為部位數
```

行(1)與行(2)宣告 SetTimer 與 KillTimer 函數以使用 Windows API 函式的計時器事件。

SetTimer 會建立一個計時器，執行在指定的逾時數值函式。這個函式需要下列參數：(1) hWnd 識別視窗以與計時器關聯。此視窗是由呼叫的執行緒所擁有。如果這個參數是 NULL，沒有視窗計時器相關聯，並忽略 nIDEvent 參數；(2) nIDEvent 指定非零的計時器識別項。如果 hWnd 參數是 NULL，這個參數會被忽略；(3) uElapse 指定逾時值，以毫秒為單位；(4)lpTimerFunc 指向函式時的逾時值經過收到通知。如果 SetTimer 函式成功，函式會傳回整數，用來識別新的計時器。KillTimer 函式需要這個要摧毀計時器的整數。如果 SetTimer 函式失敗，函式會傳回零。

建立一個計時器之後，欲取消計時器可使用 KillTimer 函式。KillTimer 函式需要下列參數： (1) hWnd 識別指定的計時器與相關的視窗。這個值必須與傳遞到 SetTimer 函式，建立計時器 hWnd 值相同；(2)uIDEvent 指定計時器會被終結。如果函式成功地終結計時器，KillTimer 函式會傳回非零值。

行(3)到行(8)作模組層次變數宣告，以讓模組內的程序或函數可以使用。

BarNo 為 K 線編號，SheetsRow 為工作表列號，Initial_Flag 為起始旗標變數。

DataRow 用以設定 BroadCast 工作表從 Dataload 工作表中取得資料之列號。

TradeNo 用以紀錄實際進出交易次數。BeginTradeNo 用以取得考慮指標回算天數後之開始交易日數。

FMA_P、SMA_P、RSI_P、RSI_B、RSI_S 分別為快速均線、慢速均線、RSI 回算 K 線數、RSI 買進參數、RSI 賣出參數。

Min_Bar 為二維陣列，第一維為 300，其為一天交易 300 秒；第二維大小為 17，分別存編號、時間、開、高、低、收、量、累積量、快速均線、慢速均線、均線決策、K 線收盤價差、RSI 指標、RSI 決策、最終決策、當 K 損益、累積損益。

PreTickTime 儲存前一 TickData 的時間。Position 為部位數。

啟動 DDE 自動交易系統程序(Start_DDE_Auto_Order)

```
1  Sub Start_DDE_Auto_Order()   '啟動離線模擬交易程序
2      Initial_Flag = 0 '設定啟始值旗標值為 0，表尚未設定
3      Position = 0 '設定起始部位為 0
4      DataRow = 1 '設定取得模擬即時資料列號
5      On Error Resume Next   '攔截錯誤語法
6      hTimer = SetTimer(0, 0, 1 * 20, AddressOf Main)   '開啟計時器事件，單位為毫秒
7      Exit Sub
8  End Sub
```

行(1)到行(8)之程序可由 OffTrading 工作表中的「開始模擬自動交易」的按鍵驅動。

其中，行(2)可設定啟始旗標(Initial_Flag)為 0，行(3) 設定起始部位為 0，行(4)設定取得模擬即時資料列號，行(5)為攔截錯誤語法。行(6)呼叫的 SetTimer 函式用以開起計時器事件；參數 3 之「1 * 20」表 1 毫秒之 20 倍，即每 0.02 秒驅動第四個參數之「Main」程序。

關閉 DDE 自動交易系統程序(**Close_DDE_Auto_Order**)

```
1  Sub Close_DDE_Auto_Order()'關閉 DDE 自動交易系統程序
2      On Error Resume Next '攔截錯誤語法
3      Call KillTimer(0, hTimer) '關閉計時器事件
4      Exit Sub
5  End Sub
```

行(1)到行(5)之程序可由 K_Bar 工作表中的「關閉 DDE 自動交易系統」的按鍵驅動，其中行(2)為攔截錯誤語法，行(3)呼叫的 KillTimer 函式用以關閉計時器事件。

主程式 Main 由 SetTimer 驅動定頻(目前為 0.02 秒)執行，其內包含讀取報價、產生定頻 K 線(目前為 1 分)，計算 MA 與 RSI 信號指標，作出信號指標多空判斷，作出策略判斷，驅動自動下單。Main 程序結構如下：

1➢變數宣告

2➢**模擬即時資料讀取**(包括時價量，將時資料由「08:45:01」轉為「0845」)

3➢**啟動啟始程序**(含技術指標參數設定與首根 K 線之首筆資料讀入)

4➢**判斷是否為新 K 線**(判斷最新即時資料之時點與上一筆即時資料時點是否相同)

4-1➢若新資料為**同一 K 線**，則產生新的最高、最低、收與量

4-2➢若新資料為**下一 K 線**，則準備輸出 K 線，計算指標，產生信號，產生策略(計算即時損益)

4-2-1➢將 K 線資料顯示於格位中並產生 K 線圖(非必要程序)

4-2-2➢計算長短均線並由均線交叉作出多空判斷

4-2-3➢計算 RSI 值並由作出多空判斷

4-2-4➢作出綜合策略判斷並由 API 送出交易單

4-2-5➢將指標計算、指標判斷、策略判斷、損益、累積損益等輸出到格位中(非必要程序)

4-2-6➢以最新「時價量」資料，**設定產生下一 K 線的起始開高低收量**

5➢**虛擬 API 下單函數**

1➢變數宣告

```
1  Private Sub Main()
2  On Error Resume Next
3  Dim I As Integer
4  Dim tmpTickTime As String, tmpTickTime1 As String, TickTime As Single
5  Dim TickPrice As Single, TickCVolume As Single, TradingCheck As String
```

行(2)為攔截錯誤語法。行(3)至(5)宣告變數。

tmpTickTime 為自 Broadcast 工作表取得的模擬報價時間(08:45:01)，tmpTickTime1 為第一次處理後的時間(084501)，TickTime 為處理後的時間(0845)。

TickPrice 為自 Broadcast 工作表取得的模擬即時成交價；TickPrice 為自 Broadcast 工作表取得的模擬即時成交量。

TradingCheck 為虛擬下單 API 函數回傳值。

2➢即時資料讀取(包括時價量，將時資料由「08:45:01」轉為「0845」)

```
1  DataRow = DataRow + 1
2  tmpTickTime = Right(Sheets("DataLoad").Cells(DataRow, 2), 8)   '時間格式為 08:45:00
3  tmpTickTime1 = Mid(tmpTickTime, 1, 2) + Mid(tmpTickTime, 4, 2) + Mid(tmpTickTime, 7, 2) '時間格式轉為 084500
4  TickTime = Int(Val(tmpTickTime1) / 100) '時間格式轉為 0845
5  TickPrice = Sheets("DataLoad").Cells(DataRow, 7)   '成交價
6  TickCVolume = Sheets("DataLoad").Cells(DataRow, 8) '成交量
```

行(1)設定依序讀取置於 DataLoad 工作表中的秒 K 線資料。

行(2)從 DataLoad 工作表之第 2 欄中依序取出即時「時間」資料。時間格式為 「08:45:00」

行(3)藉由字串擷取函數(MID)取出「時間」資料中的數值，例如「08:45:00」轉為「084500」。

行(4)將時間資料除以 100 以取得秒數值，例如將「084500」轉為「0845」。

行(5)從 DataLoad 工作表之第 7 欄中依序取得的即時「成交價」資料。

行(6)從 DataLoad 工作表之第 8 欄中依序取得的即時「成交量」資料。

3➢啟動啟始程序(含技術指標參數設定與首根 K 線之首筆資料讀入)

```
1  If Initial_Flag = 0 Then   '以 Initial_Flag 變數保證讀入
2      BarNo = 1 '設定 K Bar 編號初值
3      SheetsRow = 1 '設定列編號初值
4      TradeNo = 0 '交易次數
5      Sheets("OffTrading").Cells(2, 18) = Position '取得起始部位
6      FMA_P = Sheets("OffTrading").Cells(2, 24) '設定快速均線參數
7      SMA_P = Sheets("OffTrading").Cells(3, 24) '設定慢速均線參數
8      RSI_P = Sheets("OffTrading").Cells(2, 26) '設定 RSI 指標回算天數
9      RSI_B = Sheets("OffTrading").Cells(3, 26) '設定 RSI 指標買進指標
10     RSI_S = Sheets("OffTrading").Cells(4, 26) '設定 RSI 指標賣出指標
11     BeginTradeNo = Sheets("OffTrading").Cells(1, 28) '設定開始交易 K 線
12     PreTickTime = TickTime
13     Min_Bar(BarNo, 1) = BarNo '設定 K 線編號
14     Min_Bar(BarNo, 2) = TickTime '設定 K 線時間
15     Min_Bar(BarNo, 3) = TickPrice '設定以成交價為開盤價
16     Min_Bar(BarNo, 4) = TickPrice '設定以成交價為最高價
17     Min_Bar(BarNo, 5) = TickPrice '設定以成交價為最低價
```

```
18        Min_Bar(BarNo, 6) = TickPrice '設定以成交價為收盤價
19        Min_Bar(BarNo, 7) = TickCVolume   '設定以成交量為起始成交量
20        Initial_Flag = 1 '設定完成初始設定
21    End If
```

此段程式碼用以設定啟始程序。

行(2)設定 K Bar 編號初值，行(3)設定列編號初值，行(4)設定交易次數起始值，行(5)將起始部位設定於 OffTrading 工作表格位(R2)中。

行(6)自 OffTrading 工作表格位(X2)中取得快速均線參數，行(7)自 OffTrading 工作表格位(X3)中取得慢速均線參數，行(8)自 OffTrading 工作表格位(Z2)中取得 RSI 指標回算天數，行(9)自 OffTrading 工作表格位(Z3)中取得 RSI 指標買進值，行(10)自 OffTrading 工作表格位(Z4)中取得 RSI 指標賣出值。

行(11)設定「可下單 K 線數」(在此例中係由「長均線+1」與「RSI 計算 K 線數」兩者取其他決定，其由 OffTrading 工作表中的 AB1 格位中的「=MAX(X3+1,Z2)」決定。

行(12)將目前逐筆資料時間(TickTime)，設定至 PreTickTime 變數，以便下一逐筆資料時間讀取後可相互比較，判斷是否為同一時間區間(同一分鐘)之資料。

行(13)與行(14)分別將 K 線編號、K 線時間，設定到儲存分 K 線資料的 Min_Bar 陣列中。

行(15)～(18)，分別以最新成交價作為開盤價、最高價、最低價、收盤價，設定到儲存分 K 線資料的 Min_Bar 陣列中。

行(19)以成交量作為起始成交量，設定到儲存分 K 線資料的 Min_Bar 陣列中。

行(20)當完成初始設定，將 Initial_Flag 設為 1，則其後將不會再執行此段落程式碼。

4➢判斷是否為新 K 線(判斷最新即時資料之時點與上一筆即時資料時點是否相同)

4-1➢若新資料為同一 K 線，則產生新的最高、最低、收與量)

```
1   If PreTickTime = TickTime Then '確定為同一分鐘內之報價
2       If TickPrice > Min_Bar(BarNo, 4) Then Min_Bar(BarNo, 4) = TickPrice '產生新高價
3       If TickPrice < Min_Bar(BarNo, 5) Then Min_Bar(BarNo, 5) = TickPrice '產生新低價
4       Min_Bar(BarNo, 6) = TickPrice '產生新收價
5       Min_Bar(BarNo, 7) = Min_Bar(BarNo, 7) + TickCVolume   '以交易量相加計算成交量
```

行(1)判斷若最新的時間資料(TickTime)與前一筆時間資料相同(PreTickTime)，表示屬於同一 K 線資料，進入此段落之更新最高價、最低價、收盤價、累積量與成交量等資訊。

行(2)判斷最新的成交價若大於目前的最高價，則將最新成交價設定為目前的最高價。

行(3)判斷最新的成交價若小於目前的最低價，則將最新成交價設定為目前的最低價。

行(4)以最新成交價作為收盤價。行(5)以交易量相加計算成交量。

4-2➢若新資 為下一 K 線，則準備輸出 K 線，計算指標，產生信號，產生策略(計算即時損益)

```
    Else                                    '確定不為同一分鐘內之報價
```

以下程式碼為確定新報價與上一報價不屬同一分鐘報價後，執行指標計算、策略觸發、API 下單與準備下一分鐘 K 現起始程序等工作

4-2-1➢將 K 線資料顯示於格位中並產生 K 線圖(非必要程序)

```
1       PreTickTime = TickTime '更新 K 線時間
2       SheetsRow = SheetsRow + 1 '更新列編號值
3       '以下段落用以將開高收低量等存於 Min_Bar 陣列中資訊輸出到格位中，若欲增加執行效率可刪除
4       Sheets("OffTrading").Cells(SheetsRow, 1) = Min_Bar(BarNo, 1) '輸出 K 線編號
5       Sheets("OffTrading").Cells(SheetsRow, 2) = Min_Bar(BarNo, 2) '輸出 K 線時間
6       Sheets("OffTrading").Cells(SheetsRow, 3) = Min_Bar(BarNo, 7) '輸出 K 線成交量
7       Sheets("OffTrading").Cells(SheetsRow, 4) = Min_Bar(BarNo, 3) '輸出 K 線開盤價
8       Sheets("OffTrading").Cells(SheetsRow, 5) = Min_Bar(BarNo, 4) '輸出 K 線最高價
```

```
9         Sheets("OffTrading").Cells(SheetsRow, 6) = Min_Bar(BarNo, 5) '輸出 K 線最低價
10        Sheets("OffTrading").Cells(SheetsRow, 7) = Min_Bar(BarNo, 6) '輸出 K 線收盤價
```

行(1)用以更新 K 線時間。行(2)用以更新列編號值。

行(4)到行(10)間的程式碼用以將開高收低量等，儲存於 Min_Bar 陣列中的資訊輸出到試算表格位中。

此段程式碼輸出陣列變數值，係用以檢驗邏輯正確性，偵錯階段比較需要，若欲增加執行效率可刪除此段落。

4-2-2➢計算長短均線並由均線交叉作出多空判斷

```
1     Dim Sum As Double
2     If BarNo >= SMA_P Then '長均線值
3         Sum = 0
4         For I = BarNo - FMA_P + 1 To BarNo
5             Sum = Sum + Min_Bar(I, 6)
6         Next I
7         Min_Bar(BarNo, 9) = Sum / FMA_P
8         Sum = 0
9         For I = BarNo - SMA_P + 1 To BarNo
10            Sum = Sum + Min_Bar(I, 6)
11        Next I
12        Min_Bar(BarNo, 10) = Sum / SMA_P
13        If BarNo >= SMA_P + 1 Then
14            If (Min_Bar(BarNo - 1, 9) < Min_Bar(BarNo - 1, 10)) And (Min_Bar(BarNo, 9) >= Min_Bar(BarNo, 10)) Then   '黃金交叉判斷
15                Min_Bar(BarNo, 11) = 1
16            End If
17            If (Min_Bar(BarNo - 1, 9) > Min_Bar(BarNo - 1, 10)) And (Min_Bar(BarNo, 9) <= Min_Bar(BarNo, 10)) Then   '死交交叉判斷
18                Min_Bar(BarNo, 11) = -1
19            End If
20        End If
21    End If
```

此段程式碼以「均線交叉信號與 RSI 指標交易信號，形成交易策略」處理 K 線資料產生交易信號，並模擬 API 下單。此段落交易者可以自行決定交易策略並編碼處理 K 線交易資料。

行(2)判斷目前的 K 線資料是否足夠計算出技術指標值(必須大於等於長均線值)。

行(3)設定加總收盤價變數(Sum)的初值為 0。行(4)~行(6)以迴圈加總過去短均線計算期間的收盤價。行(7)計算均線值並存入 Min_Bar 陣列中。

行(8)到行(12)計算長均線值，原理同上。

行(13)到行(20)間的程式碼觸發均線交易信號(黃金交叉作多，死亡交叉作空)。

行(13)判斷是否進入均線交叉判斷程序(必須要「長均線值+1」才可判斷交叉)，假若長均線自第 10 日開始計算(當然，短均線早就計算了)，那麼從第 11 日才可判斷交叉。

行(14)到行(16)作黃金交叉判斷，判斷方法為「前一 K 線的短均線值(Min_Bar(BarNo - 1, 9)) < 前一 K 線的長均線值(Min_Bar(BarNo - 1, 10))」且「此一 K 線的短均線值((Min_Bar(BarNo, 9)) > 此一 K 線的長均線值(Min_Bar(BarNo, 10))」，此條件成立，則做多，並把做多行動(+1)存入 Min_Bar 陣列中。

行(17)到行(19)作死亡交叉判斷的原理相同，不再贅述。

4-2-3➢計算 RSI 值並由作出多空判斷

```
1   Dim RSI_Up_Average As Single, RSI_Up_Count As Integer, RSI_Down_Average As Single, RSI_Down_Count As Integer
2   If BarNo > 1 Then Min_Bar(BarNo, 12) = Min_Bar(BarNo, 6) - Min_Bar(BarNo - 1, 6)   '漲跌價計算
3   If BarNo >= RSI_P Then
4       RSI_Up_Average = 0: RSI_Up_Count = 0: RSI_Down_Average = 0: RSI_Down_Count = 0
5       For I = BarNo - RSI_P + 1 To BarNo
6           If Min_Bar(I, 12) > 0 Then
7               RSI_Up_Average = RSI_Up_Average + Min_Bar(I, 12)
8               RSI_Up_Count = RSI_Up_Count + 1
9           End If
10          If Min_Bar(I, 12) < 0 Then
11              RSI_Down_Average = RSI_Down_Average + Min_Bar(I, 12)
12              RSI_Down_Count = RSI_Down_Count + 1
13          End If
14      Next I
15      If RSI_Up_Count > 0 Then RSI_Up_Average = RSI_Up_Average / RSI_Up_Count Else RSI_Up_Average = 0 '計算平均漲幅
16      If RSI_Down_Count > 0 Then RSI_Down_Average = Abs(RSI_Down_Average / RSI_Down_Count) Else RSI_Down_Average = 0 '計算平均跌幅
17      If (RSI_Up_Average + RSI_Down_Average) > 0 Then '計算 RSI 指標值
18          Min_Bar(BarNo, 13) = RSI_Up_Average / (RSI_Up_Average + RSI_Down_Average)
19      Else
20          Min_Bar(BarNo, 13) = 0
21      End If
22      If Min_Bar(BarNo, 13) <= RSI_B Then Min_Bar(BarNo, 14) = 1 'RSI 指標作多
23      If Min_Bar(BarNo, 13) >= RSI_S Then Min_Bar(BarNo, 14) = -1 'RSI 指標作空
24  End If
```

此段程式碼計算 RSI 指標並觸發交易信號(RSI 值過高作空，過低作多)。

行(1)宣告 RSI 指標計算過程的變數。

行(2)判斷若 BarNo > 1，才計算此一 K 線與上一 K 線的收盤價間之漲跌差異，並存於 Min_Bar 中。

行(3)判斷若 K 線已經累積至足夠計算 RSI 值，才進入以下 RSI 計算程序。

行(4)分別設定 RSI 計算過程變數的初值。

行(5)到行(14)的迴圈分別計算 RSI 計算期間漲點總和(RSI_Up_Average)、上漲 K 線數(RSI_Up_Count)、跌點總和(RSI_Down_Average)、下跌 K 線數(RSI_Down_Count)。

行(15)與行(16)分別計算平均上漲點數((RSI_Up_Average))與平均下跌點數((RSI_Down_Average))。

行(17)到行(21)終於計算出 RSI 值。

行(22)判斷 RSI 指標過低(<=RSI_B)則作多，行(23)判斷 RSI 指標過低(<=RSI_B)則作空。

4-2-4➢作出綜合策略判斷並由 API 送出交易單

```
1   If BarNo >= BeginTradeNo Then
2       Min_Bar(BarNo, 16) = Position * (Min_Bar(BarNo, 6) - Min_Bar(BarNo, 3)) '隔 Bar 下單前計算當 Bar 損益(假設前 Bar 收=當 Bar 開)
3       Min_Bar(BarNo, 17) = Min_Bar(BarNo - 1, 17) + Min_Bar(BarNo, 16)   '計算策略累積損益
4       If Position < 1 And (Min_Bar(BarNo, 11) = 1 Or Min_Bar(BarNo, 14) = 1) Then   '當均線指標作多"且"RSI 指標作多,則作多
5           Min_Bar(BarNo, 15) = 1 '=====可以市價作多近月台指期 1 口，連接 API 送單========
```

```
6                If Position = 0 Then
7                    TradingCheck = eFOrder("TXFK0", "B", "1")   '以虛擬 API 下單接口下單
8                    Position = 1: Sheets("OffTrading").Cells(2, 18) = Position
9                End If
10               If Position = -1 Then
11                   TradingCheck = eFOrder("TXFK0", "B", "2")   '以虛擬 API 下單接口下單
12                   Position = 1: Sheets("OffTrading").Cells(2, 18) = Position
13               End If
14           End If
15           If Position > -1 And (Min_Bar(BarNo, 11) = -1 Or Min_Bar(BarNo, 14) = -1) Then
16               Min_Bar(BarNo, 15) = -1 '=====可以市價作空近月台指期 1 口，連接 API 送單========
17               If Position = 0 Then
18                   TradingCheck = eFOrder("TXFK0", "S", "1")   '以虛擬 API 下單接口下單
19                   Position = -1: Sheets("OffTrading").Cells(2, 18) = Position
20               End If
21               If Position = 1 Then
22                   TradingCheck = eFOrder("TXFK0", "S", "2")   '以虛擬 API 下單接口下單
23                   Position = -1: Sheets("OffTrading").Cells(2, 18) = Position
24               End If
25           End If
26       End If
```

行(1)判斷目前的 K 線數是否大於等於「可下單 K 線數」，以決定是否執行以下的下單判斷程序。

行(2)計算當一 K 線之損益，即若當一 K 線作多一口，且價格由 8000 點上漲到 8010 點，則賺了 10 點，反之若作空一口，則賠了 10 點(在此我們假設前一 K 線收盤價等於此一 K 線開盤價)。行(3)計算策略累積損益。當 K 損益與累積損益分別存於 Min_Bar 中。

行(4)到行(14)間的程式碼依據當**「部位(Position) < 1」且「當均線指標作多或 RSI 指標作多」，則作多**。「部位(Position) < 1」的條件係為了避免連續作多超過 1 口。**此段程式碼設定若原來部位為 0 則作多 1 口，若原來部位為 1 則作多 2 口，總之執行後會有作多 1 口部位**。當然信號組成方式與買進口數可依據需要修改。在虛擬 API 中只要使用「Call eFOrder("TXFK0", "B", "1")」(參數 1 為下單商品(現為台指期)，參數 2 設定買進("B")，參數 3 表下單 1 口("1"))就可以驅動「台指期市價買進 1 口」之 API 送單，在此的虛擬下單中，我們將下單內容送至虛擬 API 下單函數(eFOrder)中。其中，行(8)與行(12)設定下單後的部位值(Position)並將之顯示於試算表格位中。

至於行(15)到行(25)間的程式碼就是反向下單，不再贅述。

4-2-5➢將指標計算、指標判斷、策略判斷、損益、累積損益等輸出到格位中(非必要程序)

```
1        Sheets("OffTrading").Cells(SheetsRow, 9) = Format(Min_Bar(BarNo, 9), "####0.000")
2        Sheets("OffTrading").Cells(SheetsRow, 10) = Format(Min_Bar(BarNo, 10), "####0.000")
3        Sheets("OffTrading").Cells(SheetsRow, 11) = Min_Bar(BarNo, 11)
4        Sheets("OffTrading").Cells(SheetsRow, 12) = Min_Bar(BarNo, 12)
5        Sheets("OffTrading").Cells(SheetsRow, 13) = Format(Min_Bar(BarNo, 13), "0.000")
6        Sheets("OffTrading").Cells(SheetsRow, 14) = Min_Bar(BarNo, 14)
7        Sheets("OffTrading").Cells(SheetsRow, 15) = Min_Bar(BarNo, 15)
8        Sheets("OffTrading").Cells(SheetsRow, 16) = Min_Bar(BarNo, 16)
9        Sheets("OffTrading").Cells(SheetsRow, 17) = Min_Bar(BarNo, 17)
```

此段程式碼用以將指標計算過程與交易決策過程資訊輸出到格位中，若欲增加執行效率可刪除。Format 函數設定輸出格式。

4-2-6➢以最新「時價量」資料，設定產生下一 K 線的起始開高低收量

```
1        BarNo = BarNo + 1 '設定下一 K 線編號
2        Min_Bar(BarNo, 1) = BarNo '設定 K 線標號
3        Min_Bar(BarNo, 2) = TickTime '設定 K 線時間
4        Min_Bar(BarNo, 3) = TickPrice '以成交價為開盤價
5        Min_Bar(BarNo, 4) = TickPrice '以成交價為最高價
6        Min_Bar(BarNo, 5) = TickPrice '以成交價為最低價
7        Min_Bar(BarNo, 6) = TickPrice '以成交價為收盤價
8        Min_Bar(BarNo, 7) = TickCVolume '以累積量為成交量
```

此段的程式碼**用以準備下一 K 線計算**，與前述啟始程序設定的原理相同。

此後周而復始，直到收盤或按下「終止模擬交易」才會停止自動交易。

End If

End Sub

虛擬 API 下單函數

```
1   Public Function eFOrder(ByVal Symb As String, ByVal BS As String, ByVal Qty As String) As String
2   TradeNo = TradeNo + 1
3   Sheets("SimulationTradingRecord").Cells(TradeNo + 2, 1) = TradeNo
4   Sheets("SimulationTradingRecord").Cells(TradeNo + 2, 2) = Symb
5   Sheets("SimulationTradingRecord").Cells(TradeNo + 2, 3) = BS
6   Sheets("SimulationTradingRecord").Cells(TradeNo + 2, 4) = Qty
7   Sheets("SimulationTradingRecord").Cells(TradeNo + 2, 5) = Min_Bar(BarNo, 6)
8   Sheets("SimulationTradingRecord").Cells(TradeNo + 2, 6) = Min_Bar(BarNo, 17)
9   eFOrder = "1"   '傳回值 1 表交易回報成功
10  End Function
```

此函數為虛擬 API 下單函數，用以模擬期貨下單。

行(1)為函數名稱與呼叫參數，**第一個參數為「商品代號」，第二個參數為「買賣方向」，第三個參數為「買賣單位」**

行(2)累計交易單編號。

行(3)在 SimulationTradingRecord 工作表中設定交易單編號。行(4)在 SimulationTradingRecord 工作表中設定交易商品代號。行(5)在 SimulationTradingRecord 工作表中設定賣買方向。行(6)在 SimulationTradingRecord 工作表中設定交易量。行(7)在 SimulationTradingRecord 工作表中設定買賣價格(假設可以最近成交價成交)。行(8)在 SimulationTradingRecord 工作表中輸出累積損益。行(9)設定函數傳回值。

「模擬操盤系統」程式碼解析

「TradingCenter」工作表巨集

```
1   Private Sub Buy_Click()
2   If Cells(1, 12) = Cells(1, 4) - 1 Then Buy.Enabled = False
3   If Cells(1, 12) = -1 * Cells(2, 4) Then Sell.Enabled = True
4   If Cells(1, 12) < Cells(1, 4) Then Call LongEntry
5   End Sub

6   Private Sub Sell_Click()
7   If Cells(1, 12) = -1 * Cells(2, 4) + 1 Then Sell.Enabled = False
8   If Cells(1, 12) = Cells(1, 4) Then Buy.Enabled = True
9   If Cells(1, 12) > -1 * Cells(2, 4) Then Call ShortEntry
10  End Sub

11  Private Sub StartTrade_Click()
12  Dim I As Integer, J As Integer
13  '清理格位
14  Sheets("BroadCast").Range("A2:F361").ClearContents
15  Sheets("TradingRecord").Range("B3:F362").ClearContents
16  Buy.Enabled = True
17  Sell.Enabled = True
18  StopTrade.Enabled = True
19  StartTrade.Enabled = False
20  Cells(1, 12) = 0
21  Cells(2, 12) = 0
22  Cells(1, 14) = 0
23  Call StartBroadcast
24  End Sub

25  Private Sub StopTrade_Click()
26  Call CloseBroadcast
27  Buy.Enabled = False
28  Sell.Enabled = False
29  StopTrade.Enabled = False
30  StartTrade.Enabled = True
31  End Sub
```

行(1)至行(5)為工作表中「市價買進」按鍵驅動的巨集。

行(2)，若「目前部位」(L1 格位；Cells(1, 12))為「買進部位上限」(D1 格位；Cells(1, 4))減 1，則「市價買進」(Buy)按鍵驅動後，將「市價買進」按鍵(Buy)設定為無法驅動(Disable)，即 Enabled = False；因為在買進後，已達買進上限。

行(3)，若「目前部位」(L1 格位；Cells(1, 12))為「賣出部位上限」(D2 格位；-1*Cells(2, 4))，則「市價買進」(Buy)按鍵驅動後，將「市價賣出」按鍵(Sell)設定為可驅動(Enable)，即 Enabled = True，因為在買進後，已可放空。
行(4)，若「目前部位」(L1 格位；Cells(1, 12))小於「買進部位上限」(D1 格位；Cells(1, 4))，則呼叫「作多部位程序」(LongEntry)。

行(6)至行(10)為工作表中「市價賣出」按鍵驅動的巨集。
行(7)，若「目前部位」(L1 格位；Cells(1, 12))為「賣出部位上限」(D2 格位；Cells(2, 4))乘-1 加 1，則「市價賣出」(Sell)按鍵驅動後，將「市價賣出」按鍵(Sell)設定為無法驅動(Disable)，即 Enabled = False；因為在賣出後，已達賣出上限，不能再賣出。
行(8)，若「目前部位」(L1 格位；Cells(1, 12))等於「買進上限」(D1 格位；Cells(1, 4))，則「市價賣出」(Sell)按鍵驅動後，將「市價買進」按鍵(Buy)設定為可驅動(Enable)，即 Enabled = True，因為在賣出後，已可做多。
行(9)，若「目前部位」(L1 格位；Cells(1, 12))大於「賣出部位上限」(D2 格位；Cells(2, 4))，則呼叫「作空部位程序」(ShortEntry)。

行(11)至行(24)為工作表中「開始交易」(StartTrade)按鍵驅動的巨集。
行(14)清除「BroadCast」工作表「A2:F361」範圍格位。行(15)清除「TradingRecord」工作表「B3:F362」範圍格位。
行(16)、(17)與(18)，分別將「市價買進」(Buy)、「市價賣出」(Sell)與「結束交易」(StopTrade)等按鍵設定為可驅動(Enabled = True)；
行(19)將「開始交易」(StartTrade)按鍵設定為不可驅動(Enabled = False)。
行(20)、(21)與(22)，分別設定「目前部位」格位(L1 格位)為 0、「帳戶價值」格位(L2 格位)為 0、「期貨價格」格位(N1 格位)為 0
行(23)呼叫 Macro1 巨集模組中的「StartBroadcast」程序。

行(25)至行(31)為工作表中「結束交易」(StopTrade)按鍵驅動的巨集。
行(26)呼叫 Macro1 巨集模組中的「CloseBroadcast」程序。
行(27)、(28)與(29)，分別將「市價買進」(Buy)、「市價賣出」(Sell)與「結束交易」(StopTrade)等按鍵設定為不可驅動(Enabled = False)；
行(30)將「開始交易」(StartTrade)按鍵設定為可驅動(Enabled = True)。

「Macro1」巨集模組

```
1   Private Declare Function SetTimer Lib "user32" (ByVal hWnd As Long, ByVal nIDEvent As Long, ByVal uElapse As Long, ByVal
    lpTimerFunc As Long) As Long
2   Private Declare Function KillTimer Lib "user32" (ByVal hWnd As Long, ByVal nIDEvent As Long) As Long
3   Dim hTimer
4   Dim Profit As Single, U_Profit As Single, C_Profit As Single, TransNo As Integer
5   Dim BarNo As Long, Speed As Integer, TimeString As String, Price As Single, Position As Integer, TotalNetProfit As Single
6   Dim TotalTrade As Integer, LongTrade As Integer, ShortTrade As Integer, MaxProfit As Single, MinProfit As Single

7   Sub CloseBroadcast() '結束模擬交易
8       On Error Resume Next
9       Call KillTimer(0, hTimer)
10      End
11      Exit Sub
12  End Sub

13  Sub StartBroadcast()'開始模擬交易
14      BarNo = 1
```

```
15      TransNo = 0
16      MaxProfit = -1000
17      MinProfit = 1000
18      Speed = Sheets("TradingCenter").Cells(3, 2)
19      Position = Sheets("TradingCenter").Cells(1, 12)
20      On Error Resume Next
21      hTimer = SetTimer(0, 0, 1000 / Speed, AddressOf Main) '單位為毫秒
22      Exit Sub
23  End Sub

24  Sub LongEntry()'多頭進場
25      On Error Resume Next
26      Position = Position + 1: Sheets("TradingCenter").Cells(1, 12) = Position
27      Sheets("TradingCenter").Cells(2, 12) = Sheets("TradingCenter").Cells(2, 12) - 200 * Sheets("TradingCenter").Cells(1, 14)
28      TotalNetProfit = Sheets("TradingCenter").Cells(2, 12) + Sheets("TradingCenter").Cells(2, 14)
29      TransNo = TransNo + 1
30      Sheets("TradingRecord").Cells(2 + TransNo, 1) = TransNo
31      Sheets("TradingRecord").Cells(2 + TransNo, 2) = TimeString
32      Sheets("TradingRecord").Cells(2 + TransNo, 3) = 1
33      Sheets("TradingRecord").Cells(2 + TransNo, 4) = Price
34      Sheets("TradingRecord").Cells(2 + TransNo, 5) = Position
35      Sheets("TradingRecord").Cells(2 + TransNo, 6) = TotalNetProfit
36      '
37      TotalTrade = TotalTrade + 1: LongTrade = LongTrade + 1
38      If TotalNetProfit > MaxProfit Then
39          MaxProfit = TotalNetProfit
40          Sheets("TradingRecord").Cells(9, 9) = MaxProfit
41      End If
42      If TotalNetProfit < MinProfit Then
43          MinProfit = TotalNetProfit
44          Sheets("TradingRecord").Cells(10, 9) = MinProfit
45      End If
46      Sheets("TradingRecord").Cells(5, 9) = TotalTrade
47      Sheets("TradingRecord").Cells(6, 9) = LongTrade
48      Sheets("TradingRecord").Cells(8, 9) = TotalNetProfit
49      Exit Sub
50  End Sub

51  Sub ShortEntry()'空頭進場
52      On Error Resume Next
53      Position = Position - 1: Sheets("TradingCenter").Cells(1, 12) = Position
```

```
54      Sheets("TradingCenter").Cells(2, 12) = Sheets("TradingCenter").Cells(2, 12) + 200 * Sheets("TradingCenter").Cells(1, 14)
55      TotalNetProfit = Sheets("TradingCenter").Cells(2, 12) + Sheets("TradingCenter").Cells(2, 14)
56      TransNo = TransNo + 1
57      Sheets("TradingRecord").Cells(2 + TransNo, 1) = TransNo
58      Sheets("TradingRecord").Cells(2 + TransNo, 2) = TimeString
59      Sheets("TradingRecord").Cells(2 + TransNo, 3) = -1
60      Sheets("TradingRecord").Cells(2 + TransNo, 4) = Price
61      Sheets("TradingRecord").Cells(2 + TransNo, 5) = Position
62      Sheets("TradingRecord").Cells(2 + TransNo, 6) = TotalNetProfit
63      '
64      TotalTrade = TotalTrade + 1: ShortTrade = ShortTrade + 1
65      If TotalNetProfit > MaxProfit Then
66          MaxProfit = TotalNetProfit
67          Sheets("TradingRecord").Cells(9, 9) = MaxProfit
68      End If
69      If TotalNetProfit < MinProfit Then
70          MinProfit = TotalNetProfit
71          Sheets("TradingRecord").Cells(10, 9) = MinProfit
72      End If
73      Sheets("TradingRecord").Cells(5, 9) = TotalTrade
74      Sheets("TradingRecord").Cells(7, 9) = ShortTrade
75      Sheets("TradingRecord").Cells(8, 9) = TotalNetProfit
76      Exit Sub
77  End Sub

78  Private Sub Main()
79  On Error Resume Next
80  BarNo = BarNo + 1
81  Sheets("BroadCast").Cells(BarNo, 1) = Sheets("DataLoad").Cells(BarNo, 11)
82  Sheets("BroadCast").Cells(BarNo, 2) = Sheets("DataLoad").Cells(BarNo, 12)
83  Sheets("BroadCast").Cells(BarNo, 3) = Sheets("DataLoad").Cells(BarNo, 13)
84  Sheets("BroadCast").Cells(BarNo, 4) = Sheets("DataLoad").Cells(BarNo, 14)
85  Sheets("BroadCast").Cells(BarNo, 5) = Sheets("DataLoad").Cells(BarNo, 15)
86  Sheets("BroadCast").Cells(BarNo, 6) = Sheets("DataLoad").Cells(BarNo, 16)
87  Price = Sheets("BroadCast").Cells(BarNo, 6)
88  TimeString = Sheets("BroadCast").Cells(BarNo, 1)
89  Sheets("TradingCenter").Cells(1, 14) = Price
90  If BarNo > 300 Then
91      Sheets("TradingCenter").Buy.Enabled = False
92      Sheets("TradingCenter").Sell.Enabled = False
93      Sheets("TradingCenter").StopTrade.Enabled = False
```

```
94          Sheets("TradingCenter").StartTrade.Enabled = True
95          Call CloseBroadcast
96      End If
97      End Sub
```

行(1)與行(2)宣告 SetTimer 與 KillTimer 函數以使用 Windows API 函式的計時器事件。
行(3)至行(6)設定模組變數。

行(7)至行(12)間的程序用以結束模擬交易。其中行(9)呼叫 KillTimer 函數。

行(13)至行(23)間的程序用以結束模擬交易。行(21)呼叫 SetTimer 函數。行(14)、(15)、(16)、(17)分別設定播放 K 線數(BarNo)、交易次數(TransNo)、最大獲利(MaxProfit)、最低獲利(MinProfit)。
行(18)用以讀取 K 線播放速度(C2 格位)。行(19)用以讀取目前部位(L1 格位)。

行(24)至行(50)間的程式，執行多頭進場程序。
行(26)作增加部位(Position)，並將部位輸出到 L1 格位中。行(27)計算「帳戶價值」(L2 格位)。行(28)將「帳戶價值」(L2 格位)與「部位價值」(N2 格位)相加得到「淨獲利」(TotalNetProfit)。行(29)將交易次數加 1。
行(30)至行(35)分別將「交易次數」(TransNo)、「交易時間」(TimeString)、「多空口數」(1)、「交易價格」(Price)、「交易部位」(Position)、「淨獲利」(TotalNetProfit)等資料輸出到「TradingRecord」工作表 A 至 F 欄格位中。
行(37)將「總交易次數」(TotalTrade)與「做多交易次數」(LongTrade)加 1。
行(38)至行(41)間的程式碼用以判斷，若「淨獲利」(TotalNetProfit)大於「最高獲利」(MaxProfit)，則更新「最高獲利」為「淨獲利」，並於「TradingRecord」工作表 I9 格位輸出「最高獲利」值。
行(42)至行(45)間的程式碼用以判斷，若「淨獲利」(TotalNetProfit)小於「最低獲利」(MinProfit)，則更新「最低獲利」為「淨獲利」，並於「TradingRecord」工作表 I10 格位輸出「最低獲利」值。
行(46)、(47)、(48)分別將「交易次數」(TotalTrade)、「買進次數」(LongTrade)、「最後損益值」(TotalNetProfit)輸出到 I5、I6、I8 等格位。

行(51)至行(77)間的程式，執行空頭進場程序。程式設計原理，類似行(24)至行(50)間的程式執行多頭進場程序，不在贅述。

行(78)至行(97)間的程式，執行逐筆 K 線播放程序控制。
行(80)增加 K 線數。
行(81)至行(86)間程式，將「DataLoad」工作表中置於 K~P 欄中的 K 線交易時間、成交量、開盤價、最高價、最低價與收盤價，移到「DataLoad」工作表中 A ~ F 欄中以達逐筆播放 K 線交易時間、成交量、開盤價、最高價、最低價與收盤價的目標。
行(87)，讀取最新收盤價作為成交價(Price)。行(87)，讀取最新交易時間存入 TimeString 變數中。行(88)將最新成交價設定到「TradingCenter」工作表中的 N1 格位中。
由於 K 線圖設定只顯示 300 根 K 線，因此於行(90)中，判斷若 K 線數(BarNo)大於 300。
行(91)、(92)與(93)，分別將「市價買進」(Buy)、「市價賣出」(Sell)與「結束交易」(StopTrade)等按鍵設定為不可驅動(Enabled = False)；行(94)將「開始交易」(StartTrade)按鍵設定為可驅動(Enabled = True)。
行(95)呼叫「CloseBroadcast」巨集程序。

「資料讀取與轉換系統程式碼解析」程式碼解析

「Download」工作表

「載入資料」按鍵驅動程序

```
1  Private Sub SelectDate_Click()
2  Dim FileName As String '宣告存放檔名變數
3  Dim FilePath As String '宣告存放檔案路徑變數
4  Dim I As Long
5  Dim DataArray1(18000, 8) As String '宣告儲存讀入秒 K 資料的陣列
6  FilePath = ThisWorkbook.Path & "\" '取得目前活頁簿的路徑
7  '讀入開盤價資料
8  Sheets("DataLoad").Range("B2:I18001").ClearContents '清除資料讀入後寫至工作表中的範圍內容
9  FileName = Right(Str(Cells(3, 10)), 8) & ".txt" '由 Download 工作表中的 J3 格位讀入檔名(以秒 K 線日期為檔名)
10 Open FilePath & FileName For Input As #1 '開啟選取的秒 K 線檔案
11 For I = 1 To 18000 '進入讀檔程序迴圈
12     Input #1, DataArray1(I, 1), DataArray1(I, 2), DataArray1(I, 3), DataArray1(I, 4), DataArray1(I, 5), DataArray1(I, 6), DataArray1(I, 7), DataArray1(I, 8)
13     '依序讀入時間、到期日否、開盤、最高、最低、收盤、成交量、成交筆數等資料並存入 DataArray1 中
14     Cells(I + 1, 2) = " " & DataArray1(I, 1) '將日期資料寫入 B 欄中
15     Cells(I + 1, 3) = DataArray1(I, 2) '將到期日否資料寫入 C 欄中
16     Cells(I + 1, 4) = DataArray1(I, 3) '將開盤價資料寫入 D 欄中
17     Cells(I + 1, 5) = DataArray1(I, 4) '將最高價資料寫入 E 欄中
18     Cells(I + 1, 6) = DataArray1(I, 5) '將最低價資料寫入 F 欄中
19     Cells(I + 1, 7) = DataArray1(I, 6) '將收盤價資料寫入 G 欄中
20     Cells(I + 1, 8) = DataArray1(I, 7) '將成交量資料寫入 H 欄中
21     Cells(I + 1, 9) = DataArray1(I, 8) '將成交筆數資料寫入 I 欄中
22 Next I
23 Close #1 '關檔
24 End Sub
```

行(1)至行(5)宣告變數，變數目的可參考程式註解。行(6) 取得目前活頁簿的路徑。

行(8)清除資料讀入後寫至工作表中的範圍內容(B2:I18001)。

行(9)由 Download 工作表中的 J3 格位讀入檔名(以秒 K 線日期為檔名)並存於 FileName 變數中。

行(10)開啟選取的秒 K 線檔案

行(11)至(22)間為讀檔程序迴圈。

行(12) 讀入時間、到期日否、開盤、最高、最低、收盤、成交量、成交筆數等資料並存入 DataArray1 中。

行(14)至(21)將讀入資料依序放入 B 至 I 欄中。

行(23)關閉檔案。

「轉換低頻 K 線」按鍵驅動程序

```
1  Private Sub GenerateKBar_Click()
2  Dim KBar As Integer, I As Long, J As Long, NRow As Long '宣告變數
```

```
Dim TimeS As String, OpenP As Single, HighP As Single, LowP As Single, CloseP As Single, Volume As Single
Sheets("DataLoad").Range("K2:P18001").ClearContents '清除低頻 K 線寫出到工作表範圍的資料
KBar = Cells(7, 10) '從 J7 格位中讀入轉換的低頻 K 線值
NRow = 2 '設定起始寫出列號
TimeS = Right(Sheets("DataLoad").Cells(2, 2), 8) '取得首筆日期資料
Volume = Sheets("DataLoad").Cells(2, 8) '取得首筆成交量資料
OpenP = Sheets("DataLoad").Cells(2, 4) '取得首筆開盤價資料
HighP = Sheets("DataLoad").Cells(2, 5) '取得首筆最高價資料
LowP = Sheets("DataLoad").Cells(2, 6) '取得首筆最低價資料
CloseP = Sheets("DataLoad").Cells(2, 7) '取得首筆收盤價資料
For I = 2 To 18001
    If I Mod KBar <> 0 Then '判斷讀入資料是否屬於新 K 線，若否，執行 Then 後的程式
        If Sheets("DataLoad").Cells(I, 5) > HighP Then '判斷是否為更高價
            HighP = Sheets("DataLoad").Cells(I, 5) '若是，則更新最高價
        End If
        If Sheets("DataLoad").Cells(I, 6) < LowP Then '判斷是否為更低價
            LowP = Sheets("DataLoad").Cells(I, 6) '若是，則更新最低價
        End If
        Volume = Volume + Sheets("DataLoad").Cells(I, 8) '累積成交量
    Else   '判斷讀入資料是否屬於新 K 線，若是，執行 Else 後的程式
        If Sheets("DataLoad").Cells(I, 5) > HighP Then
            HighP = Sheets("DataLoad").Cells(I, 5)
        End If
        If Sheets("DataLoad").Cells(I, 6) < LowP Then
            LowP = Sheets("DataLoad").Cells(I, 6)
        End If
        Volume = Volume + Sheets("DataLoad").Cells(I, 8)
        CloseP = Sheets("DataLoad").Cells(I, 7)
        Sheets("DataLoad").Cells(NRow, 11) = TimeS
        Sheets("DataLoad").Cells(NRow, 12) = Volume
        Sheets("DataLoad").Cells(NRow, 13) = OpenP
        Sheets("DataLoad").Cells(NRow, 14) = HighP
        Sheets("DataLoad").Cells(NRow, 15) = LowP
        Sheets("DataLoad").Cells(NRow, 16) = CloseP
        NRow = NRow + 1
        '
        TimeS = Right(Sheets("DataLoad").Cells(I + 1, 2), 8)
        Volume = 0
        OpenP = Sheets("DataLoad").Cells(I + 1, 4)
        HighP = OpenP
        LowP = OpenP
```

```
44        End If
45    Next I
46    End Sub
```

行(2)至行(3)宣告變數。

行(4)清除低頻 K 線寫出到工作表範圍的資料(K2:P18001)。

行(5)從 J7 格位中讀入轉換的低頻 K 線值。

行(6)設定起始寫出列號。

行(7)至行(12)分別讀取首筆日期、成交量、開盤、最高、最低、收盤資料。

行(13)至行(45)間為處理秒 K 線資料的迴圈。

行(14)判斷讀入資料是否屬於新 K 線，若否，執行 Then 後行(15)至行(21)的程式碼。

行(15)至行(17)間程式碼判斷是否為更高價(HighP)，若是則更新最高價。行(18)至行(20)間程式碼判斷是否為更低價(LowP)，若是則更新最低價。行(21)計算累積成交量。

行(23)至(43)間的程式碼為當讀入資料屬於新 K 線時執行的程序。

行(23)至行(25)間程式碼判斷是否為更高價(HighP)，若是則更新最高價。行(26)至行(28)間程式碼判斷是否為更低價(LowP)，若是則更新最低價。行(29)計算累積成交量。行(30)以成交價為該 K 線的收盤價。

行(31)至行(36)間的程式碼輸出低頻 K 線資料。

行(37)將輸出列數加 1。

行(39)讀取下一秒 K 線時間資料作為新 K 線時間資料；行(40)重設累積交易量為 0；行(41)以下一筆秒 K 線的開盤價資料為新 K 線開盤價資料。行(42)與行(43)分別設定下一筆秒 K 線的開盤價資料為新 K 線目前的最高價資料(HighP)與最低價資料(LowP)。