

# 金融科技虛擬交易所 ( MicroTrade )

使用說明文件

昊瀚資訊股份有限公司

[HAOHANINFO.COM](http://HAOHANINFO.COM)

二〇二〇年九月十五日

## 目錄

一、MICROTRADE介面操作 .....	3
1. 登入 .....	3
2. 看盤室 .....	3
3. 交易介面 .....	5
二、PYTHON範例策略 .....	7
1. 範例策略一（固定時間進場） .....	7
2. 範例策略二（固定時間進出場） .....	7
3. 範例策略三（固定時間進出場 + 固定式停利停損） .....	7
4. 範例策略四（海龜策略） .....	8
5. 範例策略五（海龜策略 + 移動式停損） .....	9



# 一、MicroTrade 介面操作

## 1. 登入

首先至昊瀚資訊官網點選「虛擬交易所」(<https://service.haohaninfo.com/test/exchange1/VirtualExchange/login.php>)，接著輸入昊瀚資訊會員帳密（尚未申請會員帳密者請自行申請），並按下「登入」，如下圖。

還沒有昊瀚資訊帳號嗎? [註冊](#)

## 2. 看盤室

進入 MicroTrade 看盤室後，畫面上方有四種金融商品項目，分別是期貨、選擇權、股票、權證，目前僅提供期貨及股票的部份，其餘二項未來會陸續上線，如下圖。

商品名稱	成交時間	成交價	成交量	總量	買方筆數	賣方筆數	交易
MXFC1	05/21 17:19:42	12638	3	14	12	12	▼
MXFF1	05/21 17:20:12	12536	1	10	9	10	▼
MXFI0	09/15 17:23:13	12864.00	1	5252	3707	3901	▼
MXFJ0	09/15 17:23:11	12804.00	1	1627	1107	1175	▼
MXFK0	09/15 17:22:52	12770.00	1	125	101	106	▼
MXFL0	05/21 17:19:37	12734	1	31	26	24	▼

選擇金融商品項目後，畫面下方會出現該項目金融商品的即時報價，各欄位說明如下：

## 2-1. 商品名稱

- **期貨**：前三碼為商品代碼，可參考 ([https://www.haohaninfo.com/product\\_code.php](https://www.haohaninfo.com/product_code.php))，後兩碼依序為月份及年份，A 代表一月到期商品，B 代表二月到期商品，依此類推。年分則是取西元年的最後一碼。

Exp：台指 2019 年 1 月到期商品 (TXFA9)、小台指 2020 年 2 月到期商品 (MXFB0)。

- **選擇權**：前三碼為商品代碼，可參考 ([https://www.haohaninfo.com/product\\_code.php](https://www.haohaninfo.com/product_code.php))，中間五碼為履約價，後兩碼依序為月份及年分。A 代表一月到期的買權，B 代表二月到期的買權，依此類推，M 代表一月到期的賣權，N 代表二月到期的賣權，依此類推。年分則是取西元年的最後一碼。

Exp：台指選 2019 年 1 月到期履約價為 10300 點的買權商品 (TXO10300A9)、台指選 2020 年 2 月到期履約價為 10400 點的賣權商品 (TXO10400N0)。

- **股票**：如同其股票代碼。
- **權證**：如同其權證代碼。

## 2-2. 成交時間

該商品該筆 Tick 成交的時間。

## 2-3. 成交價

該商品該筆 Tick 成交的價格。

## 2-4. 成交量

該商品該筆 Tick 成交的數量。

## 2-5. 總量

該商品從開盤至當前的累計成交量。

## 2-6. 買方筆數

該商品從開盤至當前的累計買方成交筆數。

## 2-7. 賣方筆數

該商品從開盤至當前的累計賣方成交筆數。

## 2-8. 交易

按下交易按鈕可進入該商品的交易介面。

### 3. 交易介面

#### 3-1. 即時報價

畫面左上角會顯示該商品即時的成交價、成交量、買方筆數、賣方筆數、市場多空力道，前四項請參考上一節的欄位說明，市場多空力道計算公式如下：

買方平均成交口數 = 總成交量 / 總買方筆數

賣方平均成交口數 = 總成交量 / 總賣方筆數

市場多空力道 = 買方平均成交口數 / (買方平均成交口數 + 賣方平均成交口數) \* 100%

若即時報價正常，即時報價欄位右上角會顯示綠燈，異常則顯示紅燈，如下圖。



#### 3-2. 即時 K 線圖

畫面右上角會顯示該商品即時的 K 線圖，上方為成交價，下方為成交量，若委託單成交則 K 線圖會在對應的 K 棒中顯示 B (買進) 及 S (賣出)，如下圖。



### 3-3. 下單選項

畫面左下角會顯示下單選項，欄位說明如下：

- **買賣**：透過滑鼠左鍵拖移可選擇要送出委買單或委賣單。
- **委託**：可選擇送入市價單或限价單。
- **條件**：可選擇委託單的條件，市價單（MKT）搭配「立即成交否則不成交（IOC）」，限价單（LMT）搭配「當日有效單（ROD）」或「全部成交否則不成交（FOK）」。
- **數量**：委託口數或張數。
- **價格**：委託價格。按下綠色放大鏡圖示會直接顯示當前價格。

買賣	<input type="text" value="買進"/>
委託	<input type="text" value="市價單"/>
條件	<input type="text" value="立即成交否則不成交"/>
數量	<input type="text" value="1"/>
價格	<input type="text" value=""/> <input type="button" value=""/>
<input type="button" value="送出委託"/>	

### 3-4. 交易紀錄

畫面右下角會顯示交易紀錄，且 MicroTrade 網站的交易紀錄會與 GOrder 虛擬（證券／期權）的交易紀錄同步，欄位說明如下：

- **成交資訊**：已成交的委託單。
- **未成交資訊**：未成交的委託單。
- **總交易資訊**：全部的委託單，並顯示狀態。

<input checked="" type="checkbox"/> 成交資訊	<input checked="" type="checkbox"/> 未成交資訊	<input checked="" type="checkbox"/> 總交易資訊							
委託書號	商品名稱	日期	時間	價格	數量	買賣	委託類別	委託條件	委託/成交
2020091500004	MXFI0	2020-09-15	15:28:45	12000	3	S	LMT	ROD	委託
2020091500005	MXFI0	2020-09-15	15:32:33	13000	100	B	LMT	ROD	委託
2020091500004	MXFI0	2020-09-15	15:28:45	12844	3	S	LMT	ROD	成交
2020091500005	MXFI0	2020-09-15	15:32:34	12846	2	B	LMT	ROD	成交

## 二、Python 範例策略

本公司提供 Python 取報價及下單的套件，目前須搭配 GOrder 使用，詳細的使用方法可參考 GOrder 說明文件，以下僅提供 Python 即時範例策略的程式碼：

### 1. 範例策略一 ( 固定時間進場 )

#### Sample1.py

---

```
from haohaninfo import GOrder
from datetime import datetime

Broker = 'Simulator'
Prod = 'MXFC0'
StartTime = datetime.strptime('09:00:00', '%H:%M:%S')
GOQ = GOrder.GOQuote()
GOC = GOrder.GOCommand()

for i in GOQ.Subscribe(Broker, 'match', Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    if T >= StartTime:
        No = GOC.Order(Broker, Prod, 'B', '0', '1', 'IOC', 'MKT', '1')
        print(No)
    GOQ.EndSubscribe()
```

---

### 2. 範例策略二 ( 固定時間進出場 )

#### Sample2.py

---

```
from haohaninfo import GOrder
from datetime import datetime

Broker = 'Simulator'
Prod = 'MXFC0'
StartTime = datetime.strptime('09:00:00', '%H:%M:%S')
EndTime = datetime.strptime('10:00:00', '%H:%M:%S')
GOQ = GOrder.GOQuote()
GOC = GOrder.GOCommand()

for i in GOQ.Subscribe(Broker, 'match', Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    if T >= StartTime:
        No = GOC.Order(Broker, Prod, 'B', '0', '1', 'IOC', 'MKT', '1')
        print(No)
    GOQ.EndSubscribe()

for i in GOQ.Subscribe(Broker, 'match', Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    if T >= EndTime:
        No = GOC.Order(Broker, Prod, 'S', '0', '1', 'IOC', 'MKT', '1')
        print(No)
    GOQ.EndSubscribe()
```

---

### 3. 範例策略三 ( 固定時間進出場 + 固定式停利停損 )

## Sample3.py

---

```
from haohaninfo import GOrder
from datetime import datetime
import time

Broker = 'Simulator'
Prod = 'MXFC0'
TakeProfit = 20 # 停利點數
StopLoss = 20 # 停損點數
StartTime = datetime.strptime('09:00:00','%H:%M:%S')
EndTime = datetime.strptime('10:00:00','%H:%M:%S')
GOQ = GOrder.GOQuote()
GOC = GOrder.GOCommand()

# 進場判斷
for i in GOQ.Subscribe(Broker,'match',Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    if T >= StartTime:
        No = GOC.Order(Broker,Prod,'B','0','1','IOC','MKT','1')
        print(No)
        GOQ.EndSubscribe()

# 取進場價格
while True:
    time.sleep(1)
    Info = GOC.MatchAccount(Broker,No)[0].split(',')
    if Info[1] == '成交':
        OrderPrice = float(Info[4])
        break

# 出場判斷
for i in GOQ.Subscribe(Broker,'match',Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    P = float(i[2])
    Condition1 = T >= EndTime
    Condition2 = P >= OrderPrice + TakeProfit
    Condition3 = P <= OrderPrice - StopLoss
    if Condition1 or Condition2 or Condition3:
        No = GOC.Order(Broker,Prod,'S','0','1','IOC','MKT','1')
        print(No)
        GOQ.EndSubscribe()
```

---

## 4. 範例策略四 (海龜策略)

### Sample4.py

---

```
from haohaninfo import GOrder
from datetime import datetime
import time,sys

Broker = 'Simulator'
Prod = 'MXFC0'
MaxPrice = 0
MinPrice = 999999
T1 = datetime.strptime('08:45:00','%H:%M:%S')
T2 = datetime.strptime('09:00:00','%H:%M:%S')
T3 = datetime.strptime('10:00:00','%H:%M:%S')
GOQ = GOrder.GOQuote()
GOC = GOrder.GOCommand()

# 計算區間高低點
for i in GOQ.Subscribe(Broker,'match',Prod):
```



```

T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
P = float(i[2])
if T >= T1 and T < T2:
    MaxPrice = max(MaxPrice, P)
    MinPrice = min(MinPrice, P)
elif T >= T2:
    GOQ.EndSubscribe()

# 進場判斷
for i in GOQ.Subscribe(Broker, 'match', Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    P = float(i[2])
    if T < T3:
        if P > MaxPrice:
            No = GOC.Order(Broker, Prod, 'B', '0', '1', 'IOC', 'MKT', '1')
            print(No)
            GOQ.EndSubscribe()
        elif P < MinPrice:
            No = GOC.Order(Broker, Prod, 'S', '0', '1', 'IOC', 'MKT', '1')
            print(No)
            GOQ.EndSubscribe()
    else:
        sys.exit()

# 取進場方向及價格
while True:
    time.sleep(1)
    Info = GOC.MatchAccount(Broker, No)[0].split(',')
    if Info[1] == '成交':
        OrderFlag = Info[3]
        OrderPrice = float(Info[4])
        break

# 出場判斷
for i in GOQ.Subscribe(Broker, 'match', Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    if T >= T3:
        if OrderFlag == 'B':
            No = GOC.Order(Broker, Prod, 'S', '0', '1', 'IOC', 'MKT', '1')
            print(No)
            GOQ.EndSubscribe()
        elif OrderFlag == 'S':
            No = GOC.Order(Broker, Prod, 'B', '0', '1', 'IOC', 'MKT', '1')
            print(No)
            GOQ.EndSubscribe()

```

---

## 5. 範例策略五 (海龜策略 + 移動式停損)

### Sample5.py

```

from haohaninfo import GOrder
from datetime import datetime
import time, sys

Broker = 'Simulator'
Prod = 'MXFC0'
MaxPrice = 0
MinPrice = 999999
StopLoss = 20
T1 = datetime.strptime('08:45:00', '%H:%M:%S')
T2 = datetime.strptime('09:00:00', '%H:%M:%S')
T3 = datetime.strptime('10:00:00', '%H:%M:%S')
GOQ = GOrder.GOQuote()
GOC = GOrder.GOCommand()

```

```

# 計算區間高低點
for i in GOQ.Subscribe(Broker,'match',Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    P = float(i[2])
    if T >= T1 and T < T2:
        MaxPrice = max(MaxPrice,P)
        MinPrice = min(MinPrice,P)
    elif T >= T2:
        GOQ.EndSubscribe()

# 進場判斷
for i in GOQ.Subscribe(Broker,'match',Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    P = float(i[2])
    if T < T3:
        if P > MaxPrice:
            No = GOC.Order(Broker,Prod,'B','0','1','IOC','MKT','1')
            print(No)
            GOQ.EndSubscribe()
        elif P < MinPrice:
            No = GOC.Order(Broker,Prod,'S','0','1','IOC','MKT','1')
            print(No)
            GOQ.EndSubscribe()
    else:
        sys.exit()

# 取進場方向及價格
while True:
    time.sleep(1)
    Info = GOC.MatchAccount(Broker,No)[0].split(',')
    if Info[1] == '成交':
        OrderFlag = Info[3]
        OrderPrice = float(Info[4])
        break

# 出場判斷
for i in GOQ.Subscribe(Broker,'match',Prod):
    T = datetime.strptime(i[0][11:], '%H:%M:%S.%f')
    P = float(i[2])
    if OrderFlag == 'B':
        OrderPrice = max(OrderPrice,P)
        # 時間到出場
        if T >= T3:
            No = GOC.Order(Broker,Prod,'S','0','1','IOC','MKT','1')
            print(No)
            GOQ.EndSubscribe()
        # 移動式停損出場
        elif P <= OrderPrice - StopLoss:
            No = GOC.Order(Broker,Prod,'S','0','1','IOC','MKT','1')
            print(No)
            GOQ.EndSubscribe()
    elif OrderFlag == 'S':
        OrderPrice = min(OrderPrice,P)
        # 時間到出場
        if T >= T3:
            No = GOC.Order(Broker,Prod,'B','0','1','IOC','MKT','1')
            print(No)
            GOQ.EndSubscribe()
        # 移動式停損出場
        elif P >= OrderPrice + StopLoss:
            No = GOC.Order(Broker,Prod,'B','0','1','IOC','MKT','1')
            print(No)
            GOQ.EndSubscribe()

```

---