

ซอสโค้ด การดำเนินการตามขอบเขตที่กำหนดผ่านตัวชี้วัด CCI บนรูปแบบการบริหารจัดการเงิน

ในส่วนนี้จะเป็นส่วนหนึ่งของซอสโค้ดของโปรแกรมที่ใช้ในโปรแกรม Metatrader 4 ของกลุ่ม
การดำเนินการตามขอบเขตที่กำหนดผ่านตัวชี้วัด CCI บนรูปแบบการบริหารจัดการเงิน

#property copyright "KZM Type C Version.011 , The project start at 8/02/2017
Software Corp."

#property link "https://www.facebook.com/groups/239641229810857/?fref=ts"

enum MinZoneOpen{

minopen0 = 25, //25 %

minopen1 = 20, //20 %

minopen2 = 15, //15 %

minopen3 = 10, //10 %

minopen4 = 5, //5 %

minopen5 = 0, //0 %

};

enum MaxZoneOpen{

maxopen0 = 50, //50 %

maxopen1 = 55, //55 %

maxopen2 = 60, //65 %

maxopen3 = 65, //65 %

maxopen4 = 70, //70 %

maxopen5 = 75, //75 %

maxopen6 = 80, //80 %

maxopen7 = 85, //95 %

maxopen8 = 90, //90 %

maxopen9 = 95, //95 %

maxopen10 = 100, //100 %

};

```

enum Zone_Time{
    Zone_Time0 = 1, //1 Minute
    Zone_Time1 = 5, //5 Minute
    Zone_Time2 = 15, //15 Minute
    Zone_Time3 = 30, //30 Minute
    Zone_Time4 = 60, //1 Hour
    Zone_Time5 = 240, //4 Hour
    Zone_Time6 = 1440, //1 Day
    Zone_Time7 = 10080, //1 Weekly
    Zone_Time8 = 43200, //1 Monthly
};

```

```

input string _____KZM_Order_Set_1 = "Order Setting.";
input int MagicExpert = 8025;
input bool UseZoneClose = TRUE;
input string CommentOrder = "FOREX KZM";
input double Lot = 0.01;
input bool AutoMaticLot = FALSE;
input bool CommentEA = TRUE;
input int MaxOrders = 20;
input Zone_Time ZoneTime = Zone_Time4;
input Zone_Time IndicatorTime = Zone_Time2;
input string _____Range_Percent_MaxMin_Set_2 = "Percent Of Zone Order.";
input bool UseCandHighPercent = TRUE;
input MaxZoneOpen MaximumZone = maxopen0;//UseMaximumZone;
input bool UseCandLowPercent = TRUE;
input MinZoneOpen MinimumZone = minopen0;//UseMinimumZone;
input int RangeCandlestick = 0;
int RangeCandlesticks = RangeCandlestick;
input bool UseLowZone = TRUE;

```

```

input string _____Manager_Profile_Set_3 = "Management Accountant.";
input int Slippage = 1;
input string _____Min_Deposit_4 = "Good capital for deposit";
double GridPerPipW;
double GridPerPip;
input double PipDepositTest = 0.0;
double PipDeposit = PipDepositTest;
double Ori;
double cm01,cmi02;
int maxsend;
double useGrid;
double o1;
double o2;
double llo2;
double save;
double maxh, minl;

void OnInit() {
    if (RangeCandlesticks == 0) {
        RangeCandlesticks = iBars(NULL,ZoneTime);
    }
}

void OnDeinit(const int reason)
{
    ObjectDelete("nameofea");
    ObjectDelete("zonemax");
    ObjectDelete("nameofaccount");
    ObjectDelete("accmoney");
    ObjectDelete("accshowtime");
    ObjectDelete("TradingMaxZone");
}

```

```

ObjectDelete("TradingLowZone");
ObjectDelete("zonemin");
ObjectDelete("fordeposit");
ObjectDelete("forpip");
ObjectDelete("forshowgrid");
ObjectDelete("forshoworder");
ObjectDelete("forlot");
ObjectDelete("forneeddeposit");
}

int maxopen(int ai_0 = -1) {
    int n = 0;
    for (int li = OrdersTotal() - 1; li >= 0 && OrderSelect(li, SELECT_BY_POS); li--)
        if (OrderSymbol() == Symbol() && OrderMagicNumber() == MagicExpert && ai_0
== -1) {
            n++; } else
            if (ai_0 == OrderType() && OrderCloseTime() == 0) n++;
    return (n);
}

//+-----+
//| Expert tick function |
//+-----+

void OnTick()
{
    int Cand = RangeCandlesticks-1;
    double pj[500],pm[500],h1[500],minh,maxl,Zq,lgm,mizone;
    int x,j;
    double z = iHigh(Symbol(),NULL,0);
    double y = iLow(Symbol(),NULL,0);
    Ori = iClose(Symbol(),NULL,0);
    o1 = Ask;

```

```

o2 = Bid;
for(x=0;x <= Cand; x++) {
    z = iHigh(Symbol(),NULL,x);
    y = iLow(Symbol(),NULL,x);
    pj[x] = z;
    pm[x] = y;

    //Print("Bid=",z," of round=",x," in array=",pj[x]);
    //Print("Bid=",y," of round=",x," in array=",pm[x]);
}

minh = pj[0];
minl = pm[0];

for(j=0 ;j <= Cand; j++) {
    if (UseCandHighPercent) {
        if(minh > pj[j]) {
            minh = pj[j];
        } else
        if (maxh < pj[j]) {
            maxh = pj[j];
        }
        //Print("Max is:",maxh,"Min is:",minh);
    }
    if (UseCandLowPercent) {
        if(minl > pm[j]) {
            minl = pm[j];
        } else
        if (maxl < pm[j]) {
            maxl = pm[j];
        }
        //Print("Max is:",maxl,"Min is:",minl);
    }
}

```

```

}

//////////////////////////////////////////Zone
trading.////////////////////////////////
double cm;
Zq = (maxh - minl);

if (Digits == 3 || Digits == 5) {
    Zq = 10000.0 * Zq;//pip
} else {
    Zq = 1000 * Zq; //pip
}

//Print("Zone all is:",Zq," Pips");
cm = minl + (Zq / 10000); // cm = maxh
//Print("Min + Pips:",cm);
lgm = ((Zq / 100) * MaximumZone);
//Print("Zone Max is:",lgm," Pips");

if (Digits == 3 || Digits == 5) {
    cm01 = minl + (lgm / 10000);////////Zone Maximum.
    //Print("Zone Max is:",cm01);
}

mizone = ((Zq / 100) * MinimumZone);
//Print("Zone Min is:",mizone," Pips");

if (Digits == 3 || Digits == 5) {
    cmi02 = minl + (mizone / 10000);////////Zone Minimum.
    //Print("Zone Min is:",cmi02);
}

GridPerPip = (cm01 - cmi02) / MaxOrders;
useGrid = GridPerPip;

if (Digits == 3 || Digits == 5) {
    GridPerPipW = (GridPerPip / Point) / 10.0;
}

```

```

    }

    ////////////////////////////////////////Zone End.

    maxsend = maxopen();

    ////////////////////////////////////////pip

    double Mlot = Lot*10;
    if (PipDeposit == 0.0) {
        PipDeposit = (maxh - minl)/Point/10;
    }

    ////////////////////////////////////////start Cal

    double llo;
    for(int h=0 ; h <= MaxOrders; h++) {
        h1[h] = ((PipDeposit - GridPerPipW * h) * Mlot) / MarketInfo( Symbol(),
MODE_TICKVALUE);
        llo += h1[h];
    }
    llo2 = MathAbs(llo);

    ////////////////////////////////////////end cal

    if (!UseLowZone) {
        int pop = 0;
    } else
    if (UseLowZone)
    {
        pop = 1;
    }

    if (AccountBalance() >= llo2 && cmi02 > 0.0 && cm01 > 0.0 && Ori >= cmi02 *
pop && Ori <= cm01 && maxsend < MaxOrders) {

```

```

        if(iCCI(Symbol(),IndicatorTime,36,PRICE_CLOSE,0 >= 100) &&
iCCI(Symbol(),IndicatorTime,36,PRICE_CLOSE,1 < 100)) {
            if (maxsend == 0) {
                if(OrderSend( Symbol(), OP_BUY, callotsize(), Ask, Slippage *
10, 0, 0, CommentOrder + ":" + OrdersTotal(), MagicExpert, 0, Blue ))
                    maxsend++;
                Print("KZM BUY ORDER!" + OrdersTotal());
            } else if (OrderMagicNumber() == MagicExpert && maxsend > 0) {
                Grid();
            }
        }
    }
    if (maxsend > 0) {
        if (UseZoneClose && Ori > cm01) {
            ExitClose();
        } else if (!UseZoneClose) {
            ExitClose();
        }
    }
    if (CommentEA) {
        showmanycomment();
    }
}

void ExitClose() {
    int res = 0;
    if((OrderSymbol() == Symbol()) && (OrderMagicNumber() == MagicExpert)) {
        if(OrderType()==OP_BUY) {

```



```

        if (iCCI(Symbol(),IndicatorTime,36,PRICE_CLOSE,0 <= -100) &&
iCCI(Symbol(),IndicatorTime,36,PRICE_CLOSE,1 > -100) && OrderProfit() +
OrderCommission() + OrderSwap() > 0.0) {
            res = OrderClose(OrderTicket(), OrderLots(),
NormalizeDouble(Bid,Digits), Slippage * 10, Blue);
            Print("Order Closed buy!"); } }
    }
    if (save == 0) {
        save = AccountBalance() / 2;
    } else
        if (save > 0 && AccountBalance() < save) {
            res = OrderClose(OrderTicket(), OrderLots(), NormalizeDouble(Bid,Digits),
Slippage * 10, Blue);
            Print("Order Closed buy!");
        }
    }
}

void buy() {
    int ticket2;
    ticket2 = OrderSend( Symbol(), OP_BUY, callotsize(), Ask, Slippage * 10, 0, 0,
CommentOrder + ":" + OrdersTotal(), MagicExpert, 0, Blue );
    if (ticket2 > 0){
        maxsend++;
        Print("KZM BUY ORDER!" + OrdersTotal());
    } else {
        Print("ERROR SEND ORDER!!");
    }
}

void Grid() {
    if (OrderSymbol() == Symbol() && OrderComment() == CommentOrder + ":0") {

```

```

        if (o1 > OrderOpenPrice()) {
            for(int g = 1 ; g <= MaxOrders-1; g++) {
                if (maxsend == g && o1 > OrderOpenPrice() + useGrid * g) {
                    buy();
                }
            }
        } else
        if (o2 < OrderOpenPrice()) {
            for(int a = 1 ; a <= MaxOrders-1; a++) {
                if (maxsend == a && o2 < OrderOpenPrice() - useGrid * a) {
                    buy();
                }
            }
        }
    }
}

double callotsize() {
    double callot;
    if (AutoMaticLot) {
        callot = (AccountBalance() / llo2) * Lot;
    } else {
        callot = Lot;
    }
    return (callot);
}

void showmanycomment() {
    string nameofea = "nameofea", showhzone = "zonemax", accountname =
"nameofaccount", accbalace = "accmoney", acctime = "accshowtime",
MaxzoneForTrade = "TradingMaxZone";

```

```

string LowzoneForTrade = "TradingLowZone", showlzone = "zonemin", deposit =
"fordeposit", pip = "forpip", showgrid = "forshowgrid", showorders = "forshoworder",
showlot = "forlot", NeedDeposit = "forneeddeposit";

```

```

ObjectCreate(nameeea, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(nameeea, OBJPROP_XDISTANCE, 10);
ObjectSet(nameeea, OBJPROP_YDISTANCE, 20);
ObjectSetText(nameeea, "KZM v0.11 Forex ea", 13, "Broadway", Gold);

```

```

ObjectCreate(showhzone, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(showhzone, OBJPROP_XDISTANCE, 10);
ObjectSet(showhzone, OBJPROP_YDISTANCE, 39);
ObjectSetText(showhzone, " HZone : " + DoubleToStr(maxh, 5), 10, "@Arial
Unicode MS", clrAquamarine);

```

```

ObjectCreate(MaxzoneForTrade, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(MaxzoneForTrade, OBJPROP_XDISTANCE, 10);
ObjectSet(MaxzoneForTrade, OBJPROP_YDISTANCE, 52);
ObjectSetText(MaxzoneForTrade, " Max zone trading : " + DoubleToStr(cm01, 5),
10, "@Arial Unicode MS", clrAquamarine);

```

```

ObjectCreate(LowzoneForTrade, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(LowzoneForTrade, OBJPROP_XDISTANCE, 10);
ObjectSet(LowzoneForTrade, OBJPROP_YDISTANCE, 68);
ObjectSetText(LowzoneForTrade, " Min zone trading : " + DoubleToStr(cmi02, 5),
10, "@Arial Unicode MS", clrAquamarine);

```

```

ObjectCreate(showlzone, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(showlzone, OBJPROP_XDISTANCE, 10);
ObjectSet(showlzone, OBJPROP_YDISTANCE, 84);

```

```
ObjectSetText(showlzone, " Lzone : " + DoubleToStr(minl, 5), 10, "@Arial Unicode MS", clrAquaamarine);
```

```
ObjectCreate(deposit, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(deposit, OBJPROP_XDISTANCE, 10);
ObjectSet(deposit, OBJPROP_YDISTANCE, 100);
ObjectSetText(deposit, " Recommended Minimum Capital : " + DoubleToStr(llo2, 1) + "USD.", 10, "@Arial Unicode MS", clrAquaamarine);
```

```
ObjectCreate(deposit, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(deposit, OBJPROP_XDISTANCE, 10);
ObjectSet(deposit, OBJPROP_YDISTANCE, 100);
ObjectSetText(deposit, " From : " + DoubleToStr(PipDeposit,1) + "Pips.", 10, "@Arial Unicode MS", clrAquaamarine);
```

```
ObjectCreate(showgrid, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(showgrid, OBJPROP_XDISTANCE, 10);
ObjectSet(showgrid, OBJPROP_YDISTANCE, 116);
ObjectSetText(showgrid, " GridPer : " + DoubleToStr(GridPerPipW,5) + "Pips.", 10, "@Arial Unicode MS", clrAquaamarine);
```

```
ObjectCreate(accountname, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(accountname, OBJPROP_XDISTANCE, 10);
ObjectSet(accountname, OBJPROP_YDISTANCE, 132);
ObjectSetText(accountname, " Name : " + AccountName(), 10, "@Arial Unicode MS", clrAquaamarine);
```

```
ObjectCreate(accbalace, OBJ_LABEL, 0, 0, 0, 0);
ObjectSet(accbalace, OBJPROP_XDISTANCE, 10);
ObjectSet(accbalace, OBJPROP_YDISTANCE, 148);
```

```

    ObjectSetText(accbalance, " Balance : " + AccountBalance() + " "+
AccountCurrency() , 10, "@Arial Unicode MS", clrAquamarine);

```

```

    ObjectCreate(acctime, OBJ_LABEL, 0, 0, 0, 0);
    ObjectSet(acctime, OBJPROP_XDISTANCE, 10);
    ObjectSet(acctime, OBJPROP_YDISTANCE, 164);
    ObjectSetText(acctime, " Date - Time : " + TimeToStr(TimeLocal()) , 10, "@Arial
Unicode MS", clrAquamarine);

```

```

    ObjectCreate(showorders, OBJ_LABEL, 0, 0, 0, 0);
    ObjectSet(showorders, OBJPROP_XDISTANCE, 10);
    ObjectSet(showorders, OBJPROP_YDISTANCE, 180);
    ObjectSetText(showorders, " MaxOrders : " + IntegerToString(maxsend) , 10, "@Arial
Unicode MS", clrAquamarine);

```

```

    ObjectCreate(showlot, OBJ_LABEL, 0, 0, 0, 0);
    ObjectSet(showlot, OBJPROP_XDISTANCE, 10);
    ObjectSet(showlot, OBJPROP_YDISTANCE, 196);
    ObjectSetText(showlot, " Lot is : " + DoubleToStr(Lot, 2), 10, "@Arial Unicode MS",
clrAquamarine);

```

```

    ObjectCreate(NeedDeposit, OBJ_LABEL, 0, 0, 0, 0);
    ObjectSet(NeedDeposit, OBJPROP_XDISTANCE, 10);
    ObjectSet(NeedDeposit, OBJPROP_YDISTANCE, 212);
    ObjectSetText(NeedDeposit, " Minimum Capital : " + DoubleToStr(llo2, 1) + " "+
AccountCurrency(), 10, "@Arial Unicode MS", clrAquamarine);
}

```