

```

property copyright "Copyright 2021, Zyuna32246"
property link     "https://zyunafx.com/"
property version  "1.00"
property description "移動平均線のゴールデンクロスで買い、デッドクロスで売るEAです"
property strict

//---バラメーター設定
sinput int MagicNumber=2021;           //マジックナンバー
input int Slippage=10;                  //スリッページ
input double Lots=0.01;                 //ロット数
input int TakeProfit=100;                //利確幅(pips)
input int StopLoss=50;                  //損切幅(pips)
input bool TrailingSwitching=true;      //トレーリング (On=true; OFF=false)
input int TrailingPips=20;               //トレーリングストップ幅(pips)
input string comment="MA_Cross_Zn";      //コメント

//---インシケーター設定
input string MA1="";                   //移動平均線(短期)
input int Period1=20;                  //期間
input ENUM_MA_METHOD Method1=MODE_SMA; //方法
input ENUM_APPLIED_PRICE Price1=PRICE_CLOSE; //価格
input string MA2="";                   //移動平均線(長期)
input int Period2=50;                  //期間
input ENUM_MA_METHOD Method2=MODE_SMA; //方法
input ENUM_APPLIED_PRICE Price2=PRICE_CLOSE; //価格

//---プログラム全体に影響する変数
double Pips;
int SLP;

//-----
int OnInit() {                         //EA導入時に一度だけ動くプログラム
    Pips=AdjustPoint(_Symbol);
    SLP=(int)AdjustSlippage(_Symbol, Slippage);

    return(INIT_SUCCEEDED);
}

//-----
void OnTick() {                         //ティック受信する度に動くプログラム

    //トレーリングストップ
    if(TrailingSwitching)TrailingStop();

    //ポジション数を確認
    int BuyPosition=Position(1);
    int SellPosition=Position(-1);

    //エントリー判定
    int Entry=MACrossOrder();
    if(BuyPosition==0 && Entry==1)OpenOrder(1);
    if(SellPosition==0 && Entry==1)OpenOrder(-1);

    //クローズ判定
    if(BuyPosition!=0 && MACrossClose(1)==1)CloseOrder(1);
    if(SellPosition!=0 && MACrossClose(-1)==-1)CloseOrder(-1);

} //OnTick
//-----
void OnDeinit(const int reason) { //EA削除時に最後に動くプログラム
}

//-----          //オリジナル関数
//ポジション調整
double AdjustPoint(string Currency) {
    int SymbolDigits=(int)MarketInfo(Currency, MODE_DIGITS);
    double CalculatedPoint=0.0;
    if(SymbolDigits==2 || SymbolDigits==3) {
        CalculatedPoint=0.01;
    } else
    if(SymbolDigits==4 || SymbolDigits==5) {
        CalculatedPoint=0.0001;
    }
    return(CalculatedPoint);
}

//スリッページ調整
double AdjustSlippage(string Currency, int SlippagePips) {
    double CalculatedSlippage=0.0;
    int SymbolDigits=(int)MarketInfo(Currency, MODE_DIGITS);
    if(SymbolDigits==2 || SymbolDigits==3) {
        CalculatedSlippage=SlippagePips;
    } else
    if(SymbolDigits==4 || SymbolDigits==5) {
        CalculatedSlippage=SlippagePips*10;
    }
    return(CalculatedSlippage);
}

//ポジション数を取得
int Position(int PositionDirection) {
    int res=0;
    for(int i=0;i<OrdersTotal();i++) {
        if(OrderSelect(i, SELECT_BY_POS, MODE_TRADES)==true) {
            if(OrderSymbol()==_Symbol && OrderMagicNumber()==MagicNumber) {
                if(PositionDirection==1) {
                    if(OrderType()==OP_BUY)res++;
                } else
                if(PositionDirection==-1) {
                    if(OrderType()==OP_SELL)res++;
                }
            }
        }
    }
}

```

```

        return(res);
    }

//ポジションエントリー関数
void OpenOrder(int EntryPosition) {
    int res=0;
    double TP=TakeProfit*Pips;
    double SL=StopLoss*Pips;
    if(EntryPosition==1) {
        res=OrderSend(_Symbol,OP_BUY,Lots,Ask,SLP,Ask-TP,comment,MagicNumber,0,clrBlue);
    }else
    if(EntryPosition== -1) {
        res=OrderSend(_Symbol,OP_SELL,Lots,Bid,SLP,Bid+SL,Bid-TP,comment,MagicNumber,0,clrRed);
    }
}

//ポジションクローズ関数
void CloseOrder(int ClosePosition) {
    int res=0;
    for(int i=OrdersTotal()-1;i>=0;i--) {
        if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==true) {
            if(OrderSymbol() ==_Symbol && OrderMagicNumber() == MagicNumber) {
                if(OrderType() == OP_BUY && ClosePosition==1) {
                    res=OrderClose(OrderTicket(),OrderLots(),OrderClosePrice(),SLP,clrMagenta);
                }else
                if(OrderType() == OP_SELL && ClosePosition== -1) {
                    res=OrderClose(OrderTicket(),OrderLots(),OrderClosePrice(),SLP,clrMagenta);
                }
            }
        }
    }
}

//トレーリングストップ
void TrailingStop() {
    double TSP;
    int res=0;
    for(int i=OrdersTotal()-1;i>=0;i--) {
        if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==true) {
            if(OrderSymbol() == _Symbol && OrderMagicNumber() == MagicNumber) {
                if(OrderType() == OP_BUY) {
                    TSP=Ask-TrailingPips*Pips;
                    if(OrderOpenPrice()<TSP && OrderStopLoss()<TSP) {
                        res=OrderModify(OrderTicket(),OrderOpenPrice(),TSP,0,0,clrYellow);
                    }
                }else
                if(OrderType() == OP_SELL) {
                    TSP=Bid+TrailingPips*Pips;
                    if(OrderOpenPrice()>TSP && OrderStopLoss()>TSP) {
                        res=OrderModify(OrderTicket(),OrderOpenPrice(),TSP,0,0,clrYellow);
                    }
                }
            }
        }
    }
}

//移動平均線クロスエントリー
int MACrossOrder() {
    int entry=0;
    double MAS1=iMA(_Symbol,0,Period1,0,Method1,Price1,1);
    double MAS2=iMA(_Symbol,0,Period1,0,Method1,Price1,2);
    double MAL1=iMA(_Symbol,0,Period2,0,Method2,Price2,1);
    double MAL2=iMA(_Symbol,0,Period2,0,Method2,Price2,2);

    if(MAS2<=MAL2 && MAL1<MAS1)entry=1;
    if(MAS2>=MAL2 && MAL1>MAS1)entry=-1;

    return(entry);
}

//移動平均線クロスクローズ
int MACrossClose(int Position) {
    int close=0;
    double MAS1=iMA(_Symbol,0,Period1,0,Method1,Price1,1);
    double MAL1=iMA(_Symbol,0,Period2,0,Method2,Price2,1);

    if(Position==1) {
        if(MAL1>MAS1)close=1;
    }else
    if(Position== -1) {
        if(MAL1<MAS1)close=-1;
    }

    return(close);
}

```