

# MQL4 COURSE

By Coders' guru

[www.forex-tsd.com](http://www.forex-tsd.com)

-16-

Váš první Expert Advisor

Část 4

-----

**N**a cestě za poznáním jsme doputovali k poslednímu dílu části, která pojednává o Expertech (Expert advisors) – Automatizovaných obchodních systémech. Nyní pojďme společně učinit poslední krok.

## Náš stávající programový kód:

```
//+-----+
//|                                     My_First_EA.mq4 |
//|                                     Coders Guru |
//|                                     http://www.forex-tsd.com |
//+-----+
#property copyright "Coders Guru"
#property link      "http://www.forex-tsd.com"

//---- vstupní parametry
extern double TakeProfit=250.0;
extern double Lots=0.1;
extern double TrailingStop=35.0;
//+-----+
//| inicializace experta |
//+-----+
int init()
{
//----

//----
return(0);
}
//+-----+
//| deinicializace experta |
//+-----+
int deinit()
{
//----

//----
return(0);
}

int Crossed (double line1 , double line2)
{
    static int last_direction = 0;
    static int current_direction = 0;
```

```

        if(line1>line2)current_direction = 1;    //nahoru
        if(line1<line2)current_direction = 2;    //dolů
        if(current_direction != last_direction) //změna
        {
            last_direction = current_direction;
            return (last_direction);
        }
        else
        {
            return (0);
        }
    }

//+-----+
//| funkce start experta |
//+-----+
int start()
{
//-----
    int cnt, ticket, total;
    double shortEma, longEma;

    if(Bars<100)
    {
        Print("bars less than 100");
        return(0);
    }
    if(TakeProfit<10)
    {
        Print("TakeProfit less than 10");
// zkoumáme hodnotu TakeProfit
        return(0);
    }

    shortEma = iMA(NULL,0,8,0,MODE_EMA,PRICE_CLOSE,0);
    longEma = iMA(NULL,0,13,0,MODE_EMA,PRICE_CLOSE,0);

    int isCrossed = Crossed (shortEma,longEma);

    total = OrdersTotal();
    if(total < 1)
    {
        if(isCrossed == 1)
        {
            ticket=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,0,Ask+TakeProfit*Point,
            "My EA",12345,0,Green);
            if(ticket>0)
            {
                if(OrderSelect(ticket,SELECT_BY_TICKET,MODE_TRADES))
                Print("BUY order opened : ",OrderOpenPrice());
            }
            else Print("Error opening BUY order : ",GetLastError());
            return(0);
        }
        if(isCrossed == 2)
        {

```

```

        ticket=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,0,
Bid-TakeProfit*Point,"My EA",12345,0,Red);
        if(ticket>0)
        {
            if(OrderSelect(ticket,SELECT_BY_TICKET,MODE_TRADES))
Print("SELL order opened : ",OrderOpenPrice());
            }
            else Print("Error opening SELL order : ",GetLastError());
            return(0);
        }
        return(0);
    }
    for(cnt=0;cnt<total;cnt++)
    {
        OrderSelect(cnt, SELECT_BY_POS, MODE_TRADES);
        if(OrderType()<=OP_SELL && OrderSymbol()==Symbol())
        {
            if(OrderType()==OP_BUY)
// otevření pozice long
            {
// můžeme uzavřít?
                if(isCrossed == 2)
                {
                    OrderClose(OrderTicket(),OrderLots(),Bid,3,Violet);
// uzavřít pozici
                    return(0); // konec
                }
// prověřujeme trailing (posuvný) stop
                if(TrailingStop>0)
                {
                    if(Bid-OrderOpenPrice()>Point*TrailingStop)
                    {
                        if(OrderStopLoss()<Bid-Point*TrailingStop)
                        {
                            OrderModify(OrderTicket(),OrderOpenPrice(),Bid-
Point*TrailingStop,OrderTakeProfit(),0,Green);
                            return(0);
                        }
                    }
                }
            }
        }
        else
// otevření pozice short
        {
// můžeme uzavřít?
            if(isCrossed == 1)
            {
                OrderClose(OrderTicket(),OrderLots(),Ask,3,Violet);
// uzavřít pozici
                return(0); // konec
            }
// prověřujeme trailing (posuvný) stop
            if(TrailingStop>0)
            {
                if((OrderOpenPrice()-Ask)>(Point*TrailingStop))
                {

```

```

        if((OrderStopLoss() > (Ask+Point*TrailingStop)) ||
(OrderStopLoss() == 0))
        {
OrderModify(OrderTicket(), OrderOpenPrice(), Ask+Point*TrailingStop,
OrderTakeProfit(), 0, Red);
        return(0);
        }
    }
}
return(0);
}
//+-----+

```

V předcházející lekcí jsme testovali funkci *OrdersTotal*, zda má hodnotu menší než **1**, abychom otevřeli objednávky typu **Buy** nebo **Sell** v případě, že dosud nebyly žádné otevřeny. Použili jsme k tomu následující kód:

```

if(total < 1)
{
    if(isCrossed == 1)
    {
        .....
    }
    if(isCrossed == 2)
    {
        .....
    }
    return(0);
}

```

To byla procedura *Open New Order*. (otevřít novou objednávku) Dnes si prostudujeme proceduru *Modify-Close Opened Orders*. (modifikovat-uzavřít otevřené objednávky)

```

for(cnt=0; cnt<total; cnt++)
{
    ....
}

```

Ve výše uvedeném bloku jsme použili cyklus typu *for* k prověření všech dosud otevřených objednávek. Na počátku cyklu se provede přiřazení *cnt = 0* a na konci cyklu je *cnt = total* (počet všech zadaných objednávek). Každá smyčka cyklu zvýší hodnotu proměnné *cnt* o **1** (*cnt++*). Takže proměnná *cnt* obsahuje v každé smyčce cyklu pozici objednávky (0,1,2,3 etc) kterou použijeme ve funkci *OrderSelect* k výběru každé objednávky podle její pozice.

Naším dnešním úkolem je prostudovat, co se děje v srdci výše popsaného cyklu.

```

OrderSelect(cnt, SELECT_BY_POS, MODE_TRADES);
    if(OrderType()<=OP_SELL && OrderSymbol()==Symbol())
    {
        ....
    }

```

Funkce *OrderSelect* se používá k výběru otevřené nebo podmíněné objednávky dle čísla štítku nebo indexu.

Použili jsme funkci *OrderSelect* před voláním funkcí *OrderType* a *OrderSymbol*, protože pokud bychom nepoužili *OrderSelect*, funkce *OrderType* a *OrderSymbol* by nepracovaly.

**Poznámka:** Funkci *OrderSelect* musíme použít před voláním obchodních funkcí (trading functions), které nemají žádné parametry:  
*OrderMagicNumber, OrderClosePrice, OrderCloseTime, OrderOpenPrice, OrderOpenTime, OrderComment, OrderCommission, OrderExpiration, OrderLots, OrderPrint, OrderProfit, OrderStopLoss, OrderSwap, OrderSymbol, OrderTakeProfit, OrderTicket a OrderType*

Vybrali jsme mód *SELECT\_BY\_POS*, to znamená, že chceme vybrat objednávku podle indexu (pozice), nikoli podle čísla štítku.

**Poznámka:** Index první objednávky je **0**, index druhé objednávky je **1**, atd.

Použili jsme mód *MODE\_TRADES*, to znamená, že chceme vybírat z aktuálně zadaných obchodních příkazů (otevřených a podmíněných objednávek), nikoli z historie.

Funkce *OrderType* vrací typ vybrané objednávky, který může být:  
*OP\_BUY, OP\_SELL, OP\_BUYLIMIT, OP\_BUYSTOP, OP\_SELLLIMIT* nebo *OP\_SELLSTOP*

Prověřili jsme, zda typ objednávky je *roven* nebo *menší* než *OP\_SELL*. To znamená, že může nastat jeden ze dvou případů: *OP\_SELL* nebo *OP\_BUY* (protože *OP\_SELL=1* a *OP\_BUY = 0*). Udělali jsme to proto, že nechceme pracovat s podmíněnými objednávkami (pending orders).

Chceme pracovat pouze s tou objednávkou, která byla zadána v okně grafu ve kterém jsme spustili našeho Experta., takže jsme otestovali hodnotu *OrderSymbol* naší objednávky pomocí hodnoty, vrácené funkcí *Symbol*, která vrací aktuální symbol. Jestliže se hodnoty rovnají, znamená to, že pracujeme s aktuálně načteným symbolem.

Následující kód tedy pracuje s hodnotami *OrderType = OP\_SELL* nebo *OP\_BUY* a *Symbol = OrderSymbol*.

---

```

if(OrderType()==OP_BUY) // otevřena pozice long
{
    ....
}

```

Pracujeme pouze se dvěma typy objednávek, první z nich je *OP\_BUY*.

Výše uvedený kód znamená v běžné řeči toto:

*Je otevřena pozice long (Buy)? Jestliže ano, proved' tuto posloupnost příkazů....*

Podívejme se, co se stane, když bude otevřena pozice long:

---

```
if(isCrossed == 2)
{
    OrderClose(OrderTicket(), OrderLots(), Bid, 3, Violet);
    // uzavřít pozici
    return(0); // konec
}
```

Zadali jsme objednávku typu **Buy** když *shortEma* protnul *longEma* směrem **nahoru**.

Proto je logické ji uzavřít když se *shortEma* a *longEma* protnou v opačném směru (**dolů**).

Otestovali jsme tedy hodnotu proměnné *isCrossed* = 2, to znamená že protnutí v opačném směru nastalo a proto tuto objednávku uzavřeme.

Použili jsme funkci *OrderClose* k uzavření objednávky. Funkce *OrderClose* zavírá určitou otevřenou objednávku podle čísla štítku. (viz appendix 2).

Číslo štítku vybrané objednávky jsme získali pomocí funkce *OrderTicket* a zadali jsme ho jako první parametr funkce *OrderClose*.

Druhý parametr funkce *OrderClose* je *Lots* (počet lotů); použili jsme funkci *OrderLots* k získání hodnoty počtu lotů u vybrané objednávky.

Třetí parametr funkce *OrderClose* je preferovaná close price, k získání bid price vybrané objednávky jsme použili funkci *Bid*.

Čtvrtým parametrem je hodnota *slippage* – nastavili jsme ji na 3.

Pátým parametrem je barva uzavírací šipky, tu jsme nastavili na hodnotu *Violet*.

A nezapoměli jsme ukončit funkci *start* příkazem *return(0)*.

---

```
// test zda je aktivní posuvný stop (trailing stop)
if(TrailingStop>0)
{
    if(Bid-OrderOpenPrice()>Point*TrailingStop)
    {
        if(OrderStopLoss()<Bid-Point*TrailingStop)
        {
            OrderModify(OrderTicket(), OrderOpenPrice(), Bid-
                Point*TrailingStop, OrderTakeProfit(), 0, Green);
            return(0);
        }
    }
}
```

**Poznámka:** Jsme stále uvnitř bloku příkazů: *if(OrderType()==OP\_BUY)*.

Nyní použijeme techniku posuvného stopu (trailing stop) na otevřenou pozici typu **Buy** v tomto bloku příkazů.

Nejdříve jsme otestovali proměnnou *TrailingStop* zadanou uživatelem, abychom zjistili, zda je její hodnota platná (větší než **0**).

Potom jsme použili techniku posuvného stopu (trailing stop) na otevřené objednávky typu **Buy** :

Modifikujeme *stoploos* objednávky v případě, že **rozdíl** aktuální **bid** price a **open price** objednávky je **větší** než *TrailingStop* a aktuální *stoploss* je **menší** než **rozdíl** aktuální **bid** price a *TrailingStop*. (posuvného stopu)

Použili jsme funkci *OrderModify* k provedení požadované modifikace.  
Toto jsou parametry, které jsme použili s funkcí *OrderModify*:

**ticket**: Získali jsme číslo štítku aktuální objednávky pomocí funkce *OrderTicket*.

**price**: Získali jsme open price objednávky pomocí funkce *OrderOpenPrice*.

**stoploss**: A nyní pozor! Protože jsme v pozici **Buy**, nastavíme náš nový *stoploss* na hodnotu **rozdílu** aktuální **Bid** ceny a hodnoty *TrailingStop*.  
To je náš způsob posouvání *stoplossu* po každém přírůstku zisku.

**Poznámka**: Stop loss je vždy nastaven **POD** aktuální **Bid** cenu u objednávky typu **Buy** a **NAD** aktuální **Ask** cenu u objednávky typu **Sell**.

**takeprofit**: Beze změn, získali jsme aktuální hodnotu zisku (profit value) objednávky pomocí funkce *OrderTakeProfit*.

**expiration**: Nenastavili jsme datum *expirace* naší objednávky, použili jsme hodnotu **0**.

**arrow\_color**: Zůstává hodnota *Green*.

Nakonec ukončíme funkci *start*.

---

```
else // otevřena pozice short
{
....
}
```

**Poznámka**: tento *else* patří k příkazu:

```
if(OrderType()==OP_BUY) // otevřena pozice long
{
....
}
```

Prostudovali jsme případ, kdy byla objednávka typu **Buy**.

Nyní prozkoumáme práci s objednávkou typu **Sell**.

Podívejme se, co uděláme v případě, že je již otevřena objednávka typu short (Sell).

---

```
if(isCrossed == 1)
{
    OrderClose(OrderTicket(),OrderLots(),Ask,3,Violet);
    // close position
    return(0); // exit
}
```

Otevřeli jsme objednávku typu **Sell** když *shortEma* protnul *longEma* **směrem dolů**.

Tuto pozici uzavřeme až se *shortEma* a *longEma* protnou opačným směrem (**nahoru**). To se stane v případě, že proměnná *isCrossed* = **1**.

Použili jsme funkci *OrderClose* k uzavření objednávky, se stejnými parametry jako v případě uzavření objednávky typu **Buy**, s výjimkou třetího parametru - preferované close price, který má v tomto případě hodnotu **Ask** price.

Poté jsme ukončili funkci *start*.

---

```
// check for posuvný stop (trailing stop)

if(TrailingStop>0)
{
    if((OrderOpenPrice()-Ask)>Point*TrailingStop)
    {
        if((OrderStopLoss()>(Ask+Point*TrailingStop)) ||
            (OrderStopLoss()==0))
        {
            OrderModify(OrderTicket(),OrderOpenPrice(),Ask+Point*TrailingStop,
                OrderTakeProfit(),0,Red);
            return(0);
        }
    }
}
```

Nyní použijeme techniku posuvného stopu na objednávku typu **Sell** v tomto bloku kódu.

Nejdříve jsme otestovali proměnnou *TrailingStop* zadanou uživatelem, abychom zjistili, zda má nebo nemá platnou hodnotu (větší než **0**).

Poté jsme použili techniku posuvného stopu na objednávku typu **Sell**:

**Modifikujeme stoploss** objednávky pokud **rozdíl** hodnot **open** price a aktuální **ask** price je **větší** než hodnota **TrailingStop** a **aktuální stoploss** je **větší** než **součet** aktuální **ask** price a hodnoty **TrailingStop**.



Použili jsme funkci *OrderModify* k provedení žádané modifikace, se stejnými parametry jako v případě modifikace otevřené objednávky typu **Buy**, s výjimkou třetího parametru, který indikuje hodnotu našeho stoplossu:

Nastavíme náš nový *stoploss* to na hodnotu **součtu** aktuální **Ask** price a *TrailingStop*.

A pátý parametr, který v tomto případě určuje barvu šipky je *Red*.  
Poté je funkce *start* ukončena pomocí příkazu *return(0)*;

---

```
return(0);
```

Tento řádek ukončí funkci *start* ve všech ostatních případech, kdy již nejsou žádné požadavky na otevření nových pozic, ani na uzavření nebo modifikaci otevřených objednávek.

Doufám, že jste si lekci užili.  
Uvítám vaše otázky a doporučení.

**Coders' Guru**  
28-12-2005