

```

//-----+
//メイン処理
//-----+
void OnTick()
{
    int signal = 0;
    double lots = Lots;
    double take_profit = TakeProfit;
    double stop_loss = StopLoss;
    Trade = true;

    if(isExitByElapsedTime == true) ExitByBars(bars1, TimeLimitBars, Magic1);

    if(Trade == false) return;
    lots = Lots;
    take_profit = TakeProfit;
    stop_loss = StopLoss;
    signal = 0;
    if(iLow(Symbol(), PERIOD_M5, 2)+PipsToPrice(0) <= iLow(Symbol(), PERIOD_M5, 1)+PipsToPrice(0)
    && iHigh(Symbol(), PERIOD_M5, 2)+PipsToPrice(0) <= iHigh(Symbol(), PERIOD_M5, 1)+PipsToPrice(0)) signal = 1;
    if(isTimeFilter == true && TimeFilter(signal, TimeFilterStartTime1, TimeFilterEndTime1) == false) signal = 0;
    if(bars1 == Bars) signal = 0;

    if(signal != 0 && getOpenLots(Magic1) == 0) {
        if(openPosition(signal, lots, take_profit, stop_loss, Magic1)) {
            bars1 = Bars;
            if(isMail == true) SendEmail(signal);
        }
    }
    lots = Lots;
    take_profit = TakeProfit;
    stop_loss = StopLoss;
    signal = 0;
    if(iLow(Symbol(), PERIOD_M5, 2)+PipsToPrice(0) >= iLow(Symbol(), PERIOD_M5, 1)+PipsToPrice(0)
    && iHigh(Symbol(), PERIOD_M5, 2)+PipsToPrice(0) >= iHigh(Symbol(), PERIOD_M5, 1)+PipsToPrice(0)) signal = -1;
    if(isTimeFilter == true && TimeFilter(signal, TimeFilterStartTime1, TimeFilterEndTime1) == false) signal = 0;
    if(bars1 == Bars) signal = 0;

    if(signal != 0 && getOpenLots(Magic1) == 0) {
        if(openPosition(signal, lots, take_profit, stop_loss, Magic1)) {
            bars1 = Bars;
            if(isMail == true) SendEmail(signal);
        }
    }
}

```

```
}  
  
//-----+  
// | ロット数取得関数  
// |   処理: 引数に指定した単数ポジションのロット数を取得  
// |   引数: ロット数を取得するポジションのマジックナンバー, 売買種別  
// |   戻り値: ポジションのロット数  
//-----+
```