

```
//GMMMA.mq4

#property indicator_chart_window

#property indicator_buffers 12

#property indicator_color1 Aqua
#property indicator_color2 Aqua
#property indicator_color3 Aqua
#property indicator_color4 Aqua
#property indicator_color5 Aqua
#property indicator_color6 Aqua
#property indicator_color7 Magenta
#property indicator_color8 Magenta
#property indicator_color9 Magenta
#property indicator_color10 Magenta
#property indicator_color11 Magenta
#property indicator_color12 Magenta

//インジケーターバッファの宣言
double Fast_EMA_1[];
double Fast_EMA_2[];
double Fast_EMA_3[];
double Fast_EMA_4[];
double Fast_EMA_5[];
double Fast_EMA_6[];
double Slow_EMA_1[];
double Slow_EMA_2[];
double Slow_EMA_3[];
double Slow_EMA_4[];
double Slow_EMA_5[];
double Slow_EMA_6[];

//変数の宣言
extern int Fast_EMA_1_Period = 3;
extern int Fast_EMA_2_Period = 5;
extern int Fast_EMA_3_Period = 8;
extern int Fast_EMA_4_Period = 10;
extern int Fast_EMA_5_Period = 12;
extern int Fast_EMA_6_Period = 15;
extern int Slow_EMA_1_Period = 30;
extern int Slow_EMA_2_Period = 35;
extern int Slow_EMA_3_Period = 40;
extern int Slow_EMA_4_Period = 45;
extern int Slow_EMA_5_Period = 50;
extern int Slow_EMA_6_Period = 60;

extern int MA_Mode = MODE_EMA;
extern int Applied_Price = PRICE_CLOSE;

void OnInit()
{
    //インジケーターバッファのインデックス
    SetIndexBuffer(0, Fast_EMA_1);
    SetIndexBuffer(1, Fast_EMA_2);
    SetIndexBuffer(2, Fast_EMA_3);
    SetIndexBuffer(3, Fast_EMA_4);
    SetIndexBuffer(4, Fast_EMA_5);
    SetIndexBuffer(5, Fast_EMA_6);
    SetIndexBuffer(6, Slow_EMA_1);
    SetIndexBuffer(7, Slow_EMA_2);
    SetIndexBuffer(8, Slow_EMA_3);
    SetIndexBuffer(9, Slow_EMA_4);
    SetIndexBuffer(10, Slow_EMA_5);
    SetIndexBuffer(11, Slow_EMA_6);

    //インジケーターのラベル
    SetIndexLabel(0, "Fast_EMA_1");
    SetIndexLabel(1, "Fast_EMA_2");
    SetIndexLabel(2, "Fast_EMA_3");
    SetIndexLabel(3, "Fast_EMA_4");
    SetIndexLabel(4, "Fast_EMA_5");
    SetIndexLabel(5, "Fast_EMA_6");
    SetIndexLabel(6, "Slow_EMA_1");
    SetIndexLabel(7, "Slow_EMA_2");
    SetIndexLabel(8, "Slow_EMA_3");
    SetIndexLabel(9, "Slow_EMA_4");
    SetIndexLabel(10, "Slow_EMA_5");
    SetIndexLabel(11, "Slow_EMA_6");

    //インジケーターのスタイル
    SetIndexStyle(0, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(1, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(2, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(3, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(4, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(5, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(6, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(7, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(8, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(9, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(10, DRAW_LINE, STYLE_SOLID);
    SetIndexStyle(11, DRAW_LINE, STYLE_SOLID);

    //インジケーターの描画開始時点
    SetIndexDrawBegin(0, Slow_EMA_6_Period);
    SetIndexDrawBegin(1, Slow_EMA_6_Period);
    SetIndexDrawBegin(2, Slow_EMA_6_Period);
    SetIndexDrawBegin(3, Slow_EMA_6_Period);
    SetIndexDrawBegin(4, Slow_EMA_6_Period);
    SetIndexDrawBegin(5, Slow_EMA_6_Period);
    SetIndexDrawBegin(6, Slow_EMA_6_Period);
    SetIndexDrawBegin(7, Slow_EMA_6_Period);
    SetIndexDrawBegin(8, Slow_EMA_6_Period);
    SetIndexDrawBegin(9, Slow_EMA_6_Period);
    SetIndexDrawBegin(10, Slow_EMA_6_Period);
    SetIndexDrawBegin(11, Slow_EMA_6_Period);

    return;
}

int start()
{
    int limit = Bars - IndicatorCounted();

    //移動平均線の計算
    for(int i = limit - 1; i >= 0; i--)
    {
        Fast_EMA_1[i] = iMA(NULL, 0, Fast_EMA_1_Period, 0, MA_Mode, Applied_Price, i);
        Fast_EMA_2[i] = iMA(NULL, 0, Fast_EMA_2_Period, 0, MA_Mode, Applied_Price, i);
        Fast_EMA_3[i] = iMA(NULL, 0, Fast_EMA_3_Period, 0, MA_Mode, Applied_Price, i);
        Fast_EMA_4[i] = iMA(NULL, 0, Fast_EMA_4_Period, 0, MA_Mode, Applied_Price, i);
        Fast_EMA_5[i] = iMA(NULL, 0, Fast_EMA_5_Period, 0, MA_Mode, Applied_Price, i);
        Fast_EMA_6[i] = iMA(NULL, 0, Fast_EMA_6_Period, 0, MA_Mode, Applied_Price, i);
        Slow_EMA_1[i] = iMA(NULL, 0, Slow_EMA_1_Period, 0, MA_Mode, Applied_Price, i);
        Slow_EMA_2[i] = iMA(NULL, 0, Slow_EMA_2_Period, 0, MA_Mode, Applied_Price, i);
        Slow_EMA_3[i] = iMA(NULL, 0, Slow_EMA_3_Period, 0, MA_Mode, Applied_Price, i);
        Slow_EMA_4[i] = iMA(NULL, 0, Slow_EMA_4_Period, 0, MA_Mode, Applied_Price, i);
        Slow_EMA_5[i] = iMA(NULL, 0, Slow_EMA_5_Period, 0, MA_Mode, Applied_Price, i);
        Slow_EMA_6[i] = iMA(NULL, 0, Slow_EMA_6_Period, 0, MA_Mode, Applied_Price, i);
    }

    return(0);
}
```