

```
//Span Model.mq4
```

```
#property indicator_chart_window  
#property indicator_buffers 5  
#property indicator_color1 SandyBrown  
#property indicator_color2 BlueViolet  
#property indicator_color3 SandyBrown  
#property indicator_color4 BlueViolet  
#property indicator_color5 Magenta
```

#property命令を記述

```
//インジケータのバッファの宣言
```

```
double Span_A_L[];  
double Span_B_L[];  
double Span_A_H[];  
double Span_B_H[];  
double Delay[];  
double Change[];  
double Standard[];
```

インジケータバッファを宣言

```
//変数の宣言
```

```
extern int Change_Period = 9;  
extern int Standard_Period = 26;  
extern int Span_B_Period = 52;
```

変数を宣言

```
int init()
```

```
{  
    IndicatorBuffers(7);
```

```
    //インジケータのバッファのインデックス
```

```
    SetIndexBuffer(0,Span_A_L);  
    SetIndexBuffer(1,Span_B_L);  
    SetIndexBuffer(2,Span_A_H);  
    SetIndexBuffer(3,Span_B_H);  
    SetIndexBuffer(4,Delay);  
    SetIndexBuffer(5,Change);  
    SetIndexBuffer(6,Standard);
```

```
    //インジケータのラベル
```

```
    SetIndexLabel(0,"Span A Line");  
    SetIndexLabel(1,"Span B Line");  
    SetIndexLabel(2,NULL);  
    SetIndexLabel(3,NULL);  
    SetIndexLabel(4,"Delay");
```

```
    //インジケータのスタイル
```

```
    SetIndexStyle(0,DRAW_LINE,STYLE_SOLID);  
    SetIndexStyle(1,DRAW_LINE,STYLE_SOLID);  
    SetIndexStyle(2,DRAW_HISTOGRAM,STYLE_SOLID);  
    SetIndexStyle(3,DRAW_HISTOGRAM,STYLE_SOLID);  
    SetIndexStyle(4,DRAW_LINE,STYLE_SOLID);
```

```
    //遅行線の描画のシフト
```

```
    SetIndexShift(4,-Standard_Period);
```

```
    //インジケータの描画開始時点
```

```
    SetIndexDrawBegin(0,Span_B_Period);  
    SetIndexDrawBegin(1,Span_B_Period);  
    SetIndexDrawBegin(2,Span_B_Period);  
    SetIndexDrawBegin(3,Span_B_Period);  
    SetIndexDrawBegin(4,Span_B_Period);
```

```
    return(0);  
}
```

基本設定を記述

```
int start()
```

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

```
    int i = 0;
```

```
    //転換線の値の計算
```

```
    for(i = limit - 1; i >= 0; i--)
```

```
    {  
        Change[i] = (High[iHighest(NULL,0,MODE_HIGH,Change_Period,i)] +  
                    Low[iLowest(NULL,0,MODE_LOW,Change_Period,i)]) / 2;  
    }
```

```
    //基準線の値の計算
```

```
    for(i = limit - 1; i >= 0; i--)
```

```
    {  
        Standard[i] = (High[iHighest(NULL,0,MODE_HIGH,Standard_Period,i)] +  
                      Low[iLowest(NULL,0,MODE_LOW,Standard_Period,i)]) / 2;  
    }
```

```
    //雲の値の計算
```

```
    for(i = limit - 1; i >= 0; i--)
```

```
    {  
        Span_A_L[i] = (Change[i] + Standard[i]) / 2;  
        Span_B_L[i] = (High[iHighest(NULL,0,MODE_HIGH,Span_B_Period,i)] +  
                      Low[iLowest(NULL,0,MODE_LOW,Span_B_Period,i)]) / 2;  
        Span_A_H[i] = (Change[i] + Standard[i]) / 2;  
        Span_B_H[i] = (High[iHighest(NULL,0,MODE_HIGH,Span_B_Period,i)] +  
                      Low[iLowest(NULL,0,MODE_LOW,Span_B_Period,i)]) / 2;  
    }
```

```
    //遅行線の値の計算
```

```
    for(i = limit - 1; i >= 0; i--)
```

```
    {  
        Delay[i] = Close[i];  
    }
```

```
    return(0);  
}
```

具体的な処理内容を記述