

























































通过输出波形图可知每个运动周期的时序为

|  |  |
| --- | --- |
| 顺时针运动 | 逆时针运动 |
| A B  1 1  0 1  0 0  1 0 | A B  1 1  1 0  0 0  0 1 |

静止状态为11 或00, A异或B的结果为0（注1）

单片机可利用定时器中断来检测A和B的状态，中断时间在5~20ms之间均可。

当检测到A^B = 0时，表示为静止状态，记下A和B的状态。

当检测到A^B = 1时，表示有转动，读取AB的状态，如果AB是从11到01或者是从00到10则为正转，反之如果AB是从11到10或者是从00到01则为反转。

此方法比较简单可靠，可以检测转动一格又可以检测快速转动。

注1: 异或运算

异或的运算方法是一个二进制运算，C语言中用A^B表示A异或B：

1^1=0  
0^0=0  
1^0=1  
0^1=1  
两者相等为0,不等为1.

[**也许是屎上最简单的旋转编码器程序, 占用一个外中断**](http://www.ourdev.cn/thread-5343284-1-1.html)

#define ROTATE\_A PIND.2  
#define ROTATE\_B PIND.3  
  
uchar CW=0;  
uchar CCW=0;  
  
// External Interrupt 0 service routine  
interrupt [EXT\_INT0] void ext\_int0\_isr(void)  
{  
if(ROTATE\_A!=ROTATE\_B)  
    {  
        CW++;      
    }  
        else CCW++;  
}  
  
void main(void)  
{  
        if ((CW|CCW)!=0)  
            {  
            DATA=DATA+CW-CCW;  
            }  
        CW=0;  
        CCW=0;     
        delay\_ms(50);  
}  
  
  
//////////////////////////////////////////  
  
占用一个外中断,电平变化触发,CW是顺时针转过的脉冲,CCW是逆时针  
  
每50ms处理一下数值,如果想要出错,起码脉冲时间要比50ms/256短才行,手指是没可能达到这速度  
  
实际应用中感觉很爽

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
// 旋转编码器脉冲计数程序，在8051上通过外部中断0实现  
// 特别说明：适应于旋转编码器脉冲频率较低的场合(对于12MHz晶体，大致可以支持 4KHz  
//           以下的旋转编码器脉冲频率)，若旋转编码器脉冲频率更高，建议采用计数器  
//           T0和T1分别实现正反向计数，此时一般需要附加少量外部电路的配合  
// 以下代码仅供参考，由此产生的任何后果，本人不承担任何责任  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
#include "reg51.h"  
  
long counter=0;                      //旋转编码器的脉冲计数值,左转为正，右转为负  
void main()  
{  
  IT0 = 1;                                                 //外部中断0为边沿触发  
  EX0 = 1;                                                       //使能外部中断0  
  EA  = 1;                                                            //使能中断  
  while(1)  
  {  
    //                在这里写代码实现你想做的事，比如计数值的进一步处理等等等等  
  }  
}  
  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
// 延时函数delay(unsigned char dly)  
// 延时值与晶体频率有关，可以自己调整变量x的值以取得合适的单位延时值  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
void delay(unsigned char dly)  
{  
  unsigned int x;  
  while(dly--)  
  {  
    x = 1000;  
    while(x--);  
  }  
}  
  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
// 外部中断0的中断处理函数void ISR0()  
// 注意：旋转编码器的A接P3.2，B接P1.7(跟你图上一样不变)  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
void ISR0()   interrupt 0  
{  
   if((P1&0x80) == 0x80)  
   {  
     delay(10); //消抖动，延时一会，确认P1.7仍然为高，如果频率比较高，延时要缩短  
     if((P1&0x80) == 0x80)  
       counter++;  
   }  
   else  
   {  
     delay(10); //消抖动，延时一会，确认P1.7仍然为低，如果频率比较高，延时要缩短  
     if((P1&0x80) == 0x00)  
     counter--;  
   }  
}

最近用到旋转编码器，在网上找了一大堆资料，发现这篇好文章。收藏到自己的空间待用。

[ME850扩展应用]旋转编码器计数程序（飞梭旋钮功能）－数码管显示  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*                                                                  \*  
\* ME850单片机开发实验仪演示程序 -  旋转编码器计数程序              \*  
\*                                                                  \*  
\* 3位数码管显示                                                    \*  
\*                                                                  \*  
\* MCU: AT89S52   晶振：11.0592MHz                                  \*  
\*                                                                  \*  
\* 版本：V1.1 (2011/07/14)                                          \*  
\* 作者：tangyuan                   \*  
\* 网站：http://www.linjve.com /                              \*  
\*                            \*  
\* 邮箱：893@linjve.com                                       \*  
\*                                                                  \*  
\*【版权】东莞市林积为实业投资有限公司   \*  
\*【声明】此程序仅用于学习与参考，引用请注明版权和作者信息！        \*  
\*                                                                  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*                                                                  \*  
\*  功能简述： (飞梭旋钮功能)                                       \*  
\*  当旋钮顺时针旋转时，计数值增加。达到最大值255后，不再响应。     \*  
\*  当逆钮顺时针旋转时，计数值减小。达到最小值0后，不再响应。       \*  
\*  当按下旋钮时，将计数值清零（归位）。                            \*  
\*                                                                  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <reg52.h>

sbit     PINA    = P1^0;  
sbit     PINB    = P1^1;  
sbit     PIND    = P1^2;

unsigned char  display[3];  
unsigned char  code  LEDData[ ] =   
{   
   0xC0,0xF9,0xA4,0xB0,0x99,0x92,  
   0x82,0xF8,0x80,0x90,0xff  
};

char code reserve[3]\_at\_ 0x3b; //保留0x3b开始的3个字节

unsigned char  counter = 0;     //编码器脉冲计数   
unsigned char  n,shift;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ms延时子函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
void delayms(unsigned int ms)  
{  
  unsigned char k;

  while (ms--)  
  {  
    for (k = 0; k < 114; k++)  
      ;  
  }  
}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

扫描编码器子函数

在编码器引脚A为低电平期间：

编码器引脚B从0到1为正转，编码器引脚B从1到0为反转。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
void scan\_encoder(void)   
{   
   static  bit  Curr\_encoder\_b;  //定义一个变量来储存当前B信号   
   static  bit  Last\_encoder\_b;  //定义一个变量来储存上次B脚信号   
   static  bit  updata= 0;

   if( PINA && PINB)           //编码器无转动退出  
   {  
     updata = 0;   
     return;  
   }

   Last\_encoder\_b = PINB;       //记录B信号

   while(!PINA)                //等待A由低变高  
   {   
     Curr\_encoder\_b = PINB;     //记录等待期间的B信号(指当前B信号)  
     updata = 1;  
   }

   if(updata)  
   {  
   updata = 0 ;  
     if( (Last\_encoder\_b == 0)&&(Curr\_encoder\_b== 1) ) //B从0到1为正转  
     {   
   if(counter == 255)  
       return;  
       counter++;          //正转计数加  
     }   
     else if( (Last\_encoder\_b == 1)&&(Curr\_encoder\_b == 0) ) //B从1到0为反转  
     {   
       if(counter == 0)  
       return;  
       counter--;         //反转计数减  
     }   
   }  
}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
void  main(void)  
{  
   P0 = 0xff;  
   P1 = 0xff;  
   P2 = 0xff;

   T2CON = 0x00;               //设置T2CON寄存器           
   TH2 = 0xfc;                 //1ms定时  
   TL2 = 0x66;  
   ET2 = 1;                    //启用Timer2中断  
   EA = 1;                     //总中断允许  
   TR2 = 1;                    //启动定时器2  
   counter = 0;                //计数单元清零

   while(1)  
   {  
   scan\_encoder();  
     if(! PIND)                //当按下旋钮时  
     {  
       counter = 0;            //计数单元清零(归位)  
       delayms(10);  
     }  
   }  
}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Timer2中断函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
void  timer2() interrupt 5   
{   
   TR2 = 0;  
   TF2 = 0;                         //手工清中断标志    
   TH2 = 0xfc;                      //1ms定时常数  
   TL2 = 0x66;

   if(n >= 3)                       //3位数码管显示  
   {  
     n = 0;  
     shift = 0xfe;                  //送位码初值  
     P2 = 0xff;                     //关闭显示  
   }  
   else  
   {  
     display[0] = counter%10;       //个位数据  
     display[1] = (counter%100)/10; //十位数据  
     display[2] = counter/100;      //百位数据  
       
     if(display[2] == 0)  
     {  
       display[2] = 0x0a;           //百位为0,不显示  
       if(display[1] == 0)  
       display[1]  =0x0a;           //十位为0,不显示  
}

     P0 = LEDData[display[n++]];    //送段码  
     P2 = shift;                    //送位码  
     shift = (shift<<1)|0x01;       //调整位码  
   }  
   TR2 = 1;  
}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
;\*                                                                  \*  
;\* ME850单片机开发实验仪演示程序 -   旋转编码器计数程序             \*  
;\*                                                                  \*  
;\* 3位数码管显示                                                    \*  
;\*                                                                  \*  
\* 版本：V1.1 (2011/07/14)                                          \*  
\* 作者：tangyuan                   \*  
\* 网站：http://www.linjve.com /                              \*  
\*                            \*  
\* 邮箱：893@linjve.com                                       \*  
\*                                                                  \*  
\*【版权】东莞市林积为实业投资有限公司   \*  
\*【声明】此程序仅用于学习与参考，引用请注明版权和作者信息！        \*  
;\*                                                                  \*  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
;\*                                                                  \*  
;\*  功能简述： (飞梭旋钮功能)                                       \*  
;\*  当旋钮顺时针旋转时，计数值增加。达到最大值255后，不再响应。     \*  
;\*  当逆钮顺时针旋转时，计数值减小。达到最小值0后，不再响应。       \*  
;\*  当按下旋钮时，将计数值清零（归位）。                            \*  
;\*                                                                  \*  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--------------------------------------------------------------------------------  
以下为回复后可浏览的内容：

       $ INCLUDE(TIMER2.INC)

        PINA     BIT  P1.0  
        PINB     BIT  P1.1  
        PIND     BIT  P1.2

        LAST\_B   BIT  20H.0  
        CURR\_B   BIT  20H.1  
        UPDATA   BIT  20H.2

        DISPLAY  EQU  40H  
        COUNT    EQU  45H

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

        ORG  0000H  
        AJMP  MAIN  
        ORG  002BH  
        AJMP  TIMER2  
        ORG  0050H

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; 主程序

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
MAIN:  
        MOV  SP,#60H           ;设置堆栈指针  
        MOV  P0,#0FFH          ;端口初始化  
        MOV  P2,#0FFH  
        MOV  P1,#0FFH  
        MOV  COUNT,#00H        ;清计数单元

        MOV  R0,#DISPLAY       ;获得显示单元首地址  
        MOV  R1,#0FEH          ;位码初值

        MOV  T2CON,#00H        ;设置T2CON寄存器  
        MOV  TH2,#0FCH         ;1ms定时常数  
        MOV  TL2,#66H  
        SETB  ET2              ;允许Timer2中断  
        SETB  EA               ;允许总中断  
        SETB  TR2  
LOOP:  
        ACALL  ENCODER\_SCAN  
        JB  PIND,LOOP          ;判断轴向开关是否按下  
        MOV  COUNT,#00H        ;清计数单元（归位）  
        AJMP   LOOP

        RET  
;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; 编码器扫描子程序

; 在编码器引脚A为低电平期间：

; 编码器引脚B从0变到1为正转，编码器引脚B从1变到0为反转。

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
ENCODER\_SCAN:  
        MOV  P1,#0FFH  
        NOP  
        NOP  
        MOV  A,P1  
        ANL  A,#03H  
        CJNE  A,#03H,SCAN1  
        AJMP  SCAN\_END         ;旋钮没有旋转退出  
SCAN1:  
        JB  PINB,SCAN1A        ;记录B脚电平  
        CLR  LAST\_B  
        AJMP  SCAN2  
SCAN1A:  
        SETB  LAST\_B  
SCAN2:    
        JB  PINA,SCAN3  
        SETB  UPDATA           ;当旋钮旋转时，A脚有低电平产生  
        JB  PINB,SCAN2A        ;在A为低电平期间，循环记录B脚电平  
        CLR  CURR\_B  
        AJMP  SCAN2  
SCAN2A:  
        SETB  CURR\_B   
        AJMP  SCAN2  
SCAN3:  
        JNB  UPDATA,SCAN\_END   ;旋钮没有旋转(没有新数据产生)退出  
        JNB  CURR\_B,SCAN5      ;  
        JB  LAST\_B,SCAN\_END  
        MOV  A,COUNT  
        CJNE A,#0FFH,SCAN4     ;判断是否达到最大值  
        AJMP  SCAN\_END  
SCAN4:  
        INC  COUNT             ;顺时针旋转，计数值加1  
        AJMP  SCAN\_END  
SCAN5:  
        JNB  LAST\_B,SCAN\_END  
        MOV  A,COUNT  
        CJNE A,#00H,SCAN6      ;判断是否达到最小值  
        AJMP  SCAN\_END  
SCAN6:  
        DEC  COUNT             ;逆时针旋转计数值减1  
SCAN\_END:  
        CLR  UPDATA            ;清更新数据标志  
        RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Timer2 中断服务子程序

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
TIMER2:  
        CLR  TR2  
        PUSH  ACC  
        PUSH  B  
        CLR  TF2               ;清中断标志  
        MOV  TH2,#0F8H         ;2ms定时常数  
        MOV  TL2,#0CCH

        MOV  A,R1  
        JB ACC.3,T2\_1          ;判3位数码管是否显示完？  
        MOV  R0,#DISPLAY       ;获得显示单元首地址  
        MOV  R1,#0FEH          ;位码初值  
        MOV  P2,#0FFH          ;关闭显示  
        AJMP  T2\_END  
T2\_1:  
        MOV A,COUNT            ;转换计数值数据  
        MOV B,#100  
        DIV AB  
        MOV DISPLAY+2,A        ;百位数据  
        XCH  A,B               ;数据交换  
        MOV  B,#10  
        DIV AB  
        MOV  DISPLAY+1,A       ;十位数据  
        MOV  DISPLAY,B         ;个位数据

        MOV  A,DISPLAY+2       ;百位数据  
        CJNE A,#00H,T2\_2  
        MOV  DISPLAY+2,#0AH    ;百位数据为0，不显示  
        MOV  A,DISPLAY+1       ;十位数据  
        CJNE A,#00H,T2\_2  
        MOV  DISPLAY+1,#0AH    ;十位数据为0，不显示  
T2\_2:  
        MOV  A,@R0             ;取显示数据  
        MOV  DPTR,#TABLE       ;获得表头  
        MOVC A,@A+DPTR         ;查表获得段码显示数据  
        MOV  P0,A              ;送段码  
        MOV  P2,R1             ;送位码  
        MOV  A,R1              ;准备下一位显示的位码  
        RL A  
        MOV  R1,A                
        INC  R0                ;取下一个显示单元地址  
T2\_END:  
        POP  B  
        POP  ACC  
        SETB  TR2  
        RETI

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TABLE:  
      DB 0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H  
      DB 0FFH  ; '熄灭'

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

        END         ;结束

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*