Intelligent Light and Smart Fan

```
**************************************************************************
           PROGRAM FOR AUTOMATIC LIGHT CONTROL AND FAN
**************************************************************************
```

```c
#include<reg52.h>
#include<stdio.h>

sbit IR_carrier  = P1^0;
sbit IR_Modulator = P1^1;
sbit light_sensor_1 = P1^2;                    //    yellow
sbit light_sensor_2 = P1^3;                    //    green
sbit bulb= P1^4;
sbit Fan_speed_H  = P1^5;
sbit Fan_speed_M  = P1^6;
sbit Fan_speed_L  = P1^7;
sbit light_T1  = P2^4;
sbit fan_key  =  P3^2;
sbit sensor_1  = P3^1;
sbit sensor_2  = P3^0;
sbit ADC_INT   = P2^0;
sbit ADC_INT  = P2^3;
sbit ADC_RD  = P2^2;
sbit ADC_WR  = P2^1;

sfr T2MOD  = 0xC9;

unsigned char count;
int left, right, update;
int IR_count  = 0;
int t2count  = 0;
int time_count;
int time_start;
int I;
int fan_count;
int key_press;
int fan_time;

int time=0;
unsigned char scan_present;
unsigned char scan_2nd;
unsigned char ADC_DATA;
unsigned char flag;
unsigned char temp;
unsigned char  DATA;
```

```
int person_count  =0;
int person_out=0;
int person_in=0;
```

```
*****************************************************************************
1. Delay Function starts
*****************************************************************************
        time1ms()
        {
        int I;
        for(i=0;i<50;i++);
        }
        void delay(int n)
        {
        int k;
                for(k=0;k<n;k++)
                time1ms();
        }

/****************************************************************************
Delay function Ends
*****************************************************************************

/****************************************************************************
2. IR function starts
*****************************************************************************
        void IR_carrier()
        {       RCAP2H=0xFF;                    //value to produce 38 KHz at P1.0 with 50% DC
                RCAP2L=0x02;
                T2MOD = 0x02;
                C_T2=0;
                TR2 =1;                         //Start Timer2
        }

        void IR_modulation()
        {
                TH1=0xD0; //0xFC;
                TL1 = 0x11; //0xCF;
                TR1 = 1;
        }

        void IR_active()
        {
                IR_carrier();
                IR_modulation();
```

```
        }

        /*
        void IR_deactive()
                {  TR0=0;
                IR_Modulator=0;
}

*****************************************************************************
IR function ends
*****************************************************************************

        void light_activation()
        {       delay(50);
                if(person_count>=1)
                        bulb=1;

                if(person_count==0)
                        bulb=0;

                        left=0;
                        right=0;
        }

*****************************************************************************
4. Scanning function starts
*****************************************************************************

        void scan_sensor_new()
        {

                        P3=0xff;
                        delay(5);
                        scan_present=P3;
                        scan_present= scan+present & 0x03;

                        if(scan_present==0x03)
                                        {
                                                light_sensor_1=0;
                                                light_sensor_2=0;
                                        }

        if(scan_present==0x02)
        {       light_sensor_1=1;
                light-sensor_2=0;
                time_start=time;

         for(i=0;i<500;i++)
             {     delay(5);
```

```c
                    scan_2nd=P3
            scan_2nd==scan_2nd & 0x03;

        if(scan_2nd==0x01)

                    {           person_count++;
                                light_activation();
                                break;
                        }
                    }
                }
        if(scan_present==0x01)
                    {
                                light_sensor_1=0;
                                light_sensor_2=1;
                                time_start=time;

                for(i=0;i<500;i++)
                        {       delay(5);
                                scan_2nd=P3;
                                scan_2nd=scan+2nd & 0x03;

                if(scan_2nd==0x02)
                        {
                                if(person_count==0)
                                {
                                        person_count=0;
                                        light_activation();
                                        break;
                                }
                                else
                                {       person_count--;
                                        light_activation();
                                        break;
                                }
                        }
                    }
            }
        if(scan_present==0x00)
        {
                light_sensor_1=1;
                        light_sensor_2=1;
                }
        }
```

****

Scanning function ends

****

```
****************************************************************
Processing function starts
****************************************************************

void fan()

        {

                fan_count++;
                if(fan_count==5)
                fan_count=0;
                if(fan_count==0)
                {
                //auto
                }

                if(fan_count==1)
                {
                        Fan_speed_H=0;
                        Fan_speed_M=0;
                        Fan_speed_L=0;
                }
                if(fan_count==1)
                {
                        Fan_speed_H=0;
                        Fan_speed_M=0;
                        Fan_speed_L=1;
                }
                if(fan_count==1)
                {
                        Fan_speed_H=0;
                        Fan_speed_M=1;
                        Fan_speed_L=0;
                }
                if(fan_count==1)
                {
                        Fan_speed_H=1;
                        Fan_speed_M=0;
                        Fan_speed_L=0;
                }

                delay(250);

                }

        void ext_fan() interrupt2
        {
                IE=0xAA;
                flag=1;
```

```c
        fan();
        IE=0xAF;

void ext_sensor() interrupt 0;
}
        ;
}

void correction()
{       DATA=0x00;

    temp=ADC_DATA;
        if(temp & 0x01) == 0x01)
        DATA= DATA + 0x80;
        temp=ADC_DATA;
        if(temp & 0x02) == 0x02)
        DATA= DATA + 0x40;

        temp=ADC_DATA;
        if(temp & 0x04) == 0x04)
        DATA= DATA + 0x20;

        temp=ADC_DATA;
        if(temp & 0x08) == 0x08)
        DATA= DATA + 0x10;

        temp=ADC_DATA;
        if(temp & 0x10) == 0x10)
        DATA= DATA + 0x08;

        temp=ADC_DATA;
        if(temp & 0x20) == 0x20)
        DATA= DATA + 0x04;

        temp=ADC_DATA;
        if(temp & 0x40) == 0x40)
        DATA= DATA + 0x80;

        temp=ADC_DATA;
        if(temp & 0x80) == 0x80)
        DATA= DATA + 0x01;

        ADC_DATA = DATA;
}

void fan_speed_auto()
{
        correction();
```

```c
        if(0<ADC_DATA)&(ADC_DATA<=0x17))
                {       Fan_speed_H             = 0;
                        Fan_speed_M           = 0;
                        Fan_speed_L         = 0;
                }
        if(0<ADC_DATA)&(ADC_DATA<=0x1A))
                {       Fan_speed_H                 = 0;
                        Fan_speed_M               = 0;
                        Fan_speed_L       = 1;
                }
        if(0<ADC_DATA)&(ADC_DATA<=0x1D))
                {       Fan_speed_H           = 0;
                        Fan_speed_M         = 1 ;
                        Fan_speed_L         = 0;
                }
        if(0<ADC_DATA)&(ADC_DATA<=0xFF))
                {       Fan_speed_H           = 1;
                        Fan_speed_M           = 0;
                        Fan_speed_L           = 0;
                }

void room_temp()
{
        ADC_INT=1;      //initialise input port
        P0=0xff;            // initialise data input port
        ADC_CS=0;        // enable the ADC
        ADC_WR=0;
        delay(10);
        ADC_WR=1;
        while(ADC_INT==1)
        {
                ;
        }

        ADC_RD=1;
        delay(10);
        ADC_RD=0;

        ADC_DATA=P);

        fan_speed_auto();
}

void fan_process_auto()

{
        if(fan_count==0;
        room_temp();
}
```

```
/*******************************************************************************
Processing function ends
*******************************************************************************/
/*

/*******************************************************************************
3.Timer Function Starts
*******************************************************************************/

        void System_clock()
        {
         //5ms increment till 1000ms using Timer1 in 16 bit mode

                TH0=0xED;
                TH0=0xFE;
                TR0=1;
        }
        void timer0() interrupt 1
        {
                time=time+5;

                if(time>=1000)
                        {       time=0;
                                light_T1=~light_T1;
                                fan_time++;
                        }

                if(fan_time==60)
                        {fan+time=0;
                                fan_process_auto();
                        }

                        System_clock();
                        }

        void timer1() interrupt3
        {               t2 count++;
                        if(t2count==100)
                                {       //light_T1=~light_T1;
                                        t2count=0;
                                }
                        IR_Modulator=~IR_Modulator;
                        IR_Modulation();
        }

/*******************************************************************************
Timer function ends
*******************************************************************************/
```

```
/*********************************************************************
5.Processing function starts
*********************************************************************/

        void process_init()
        {       P1=0;                           //Initialise P3 as input port.
                P3=0xff;                        //Initial condition of the light switch off
                light_sensor_1=0;
                light_sensor_2=0;
                TMOD=0x11;                      //Timer 1 and Timer0 as mode 1 16 bit mode
                IE=0xAF;                        // enable timer 2 and Timer 0 int
        .ext &ext0
                TT1=1;
                IT0=1;                           //edge trigger int1;
                IT1=1;
                System_clock();                  //function to count time from 0 to 1000ms with
        50ms
                IR_active();
                fan_count=0;
                fan_key=1;
                fan_process_auto();
        }
/*********************************************************************
6.Main Function starts
*********************************************************************/

        void main()
        {       process_init();
                while(1)
                {
                        scan_sensor_new();
                }
        }
```