

A laser based system that turns lights ON and OFF when a person enters or leaves the room.

Preface

Automation has always fascinated human species, now-a-days we want machines to do everything for us. This is achieved either for comfort and ease or for attaining precision. Human is good and blessed with indescribable intelligence, logical deduction and reasoning. For our ease we have programmed machines for logic, which are good in doing things repeatedly with the same precision.

I too have been fascinated by this automation since childhood. The very first thing that I got to use for automation was computer, so I learnt programming and got a taste of power it can offer. Till date I'm in love with this machine and am still getting a feel of that power. However I wanted to get this same power outside the computer as well. These things remain a blackbox for me up-till university, when in an interview of TUSSAT I learnt about the term microcontroller. As soon as I learnt it, I started exploring it up to the fullest.

This is my very first project, and was a childhood dream. It really looks cool when automatically switching off and switching on lights in a room when a person enters or leaves the room. This was the primary driving force of the project. Yet there is a lot of its application in energy efficient buildings and is probably the next thing in future that we are going to find everywhere.

One thing that I would like to mention here is that my primary aim was not to make the device very accurate, energy efficient and launch as a product, but to enjoy the beauty of automation that had fascinated my mind since childhood so I didn't give any importance to other various possible better ways of doing this. I did not copy the idea/code or style from anywhere simply because the tools that I used were taken from one of the robot (Line follower) that I got in a robotics workshop. So I just implemented what I thought myself and made it work myself.

Now in 2nd year when I'm compiling the report I found there are probably other better ways of doing the same thing that I did. Like you could avoid lasers as they consume more energy (you have to keep them ON for monitoring) or do other variety of things. You could also avoid microcontrollers as they are costly and they have a bit more power that required for the application. Moreover they require more battery power and all that stuff.

Still the application is very efficient in terms of accuracy. I tested it in my room when I was in 1st year and it worked really well. As always it appeared easy as per initial plans but later when I get to coding and testing there were a plenty of stuff. I would like to discuss all the problems that I faced during the developing, implementation and testing in this report.

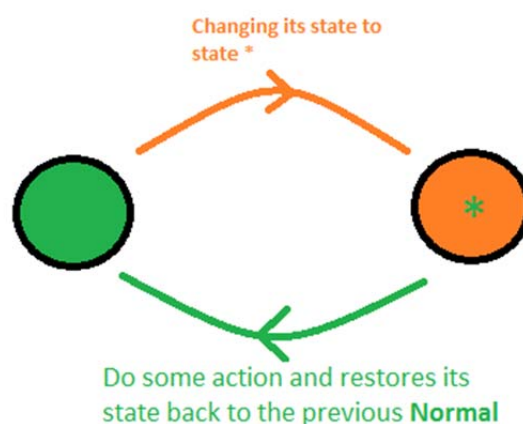
Introduction

When you have to detect the person entering you should give some information to the MCU. This information should be provided using digital logic ie 1 or 0. So if we have a laser at the door of a room, when the person entering the room cuts the laser, the sensor attached to MCU will stop receiving the signal and hence communicate a 0 signal to MCU for a given time. The MCU will interpret this signal and can determine that a person has passed in front of the laser.

However this model has one problem, the MCU will never know whether the person passed inside or outside. It just knows that a person has passed in one way. It's a simple concept that one signal can only communicate one information and in this case the information is that a person has passed whether inside or outside, it is not known.

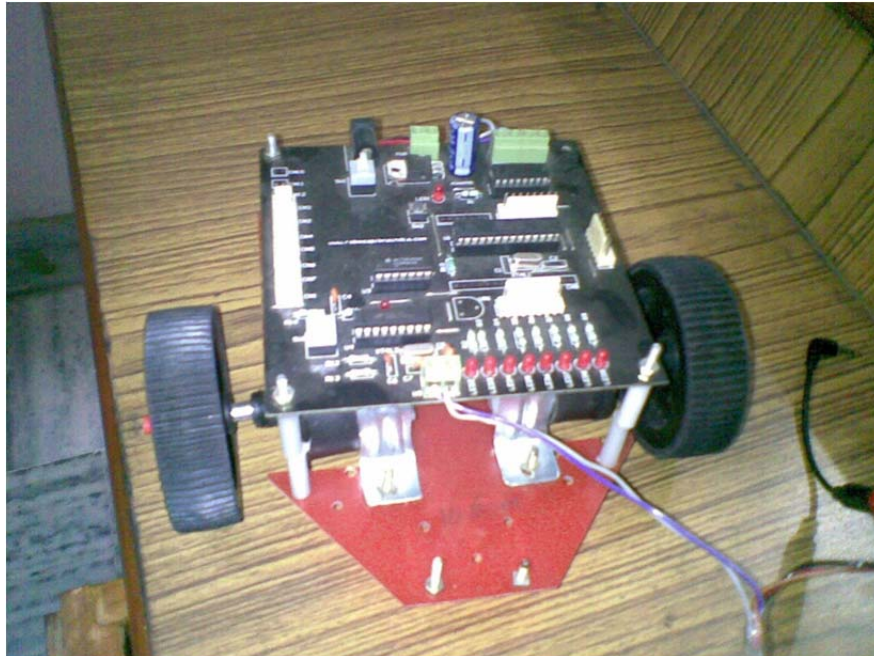
This problem can be easily tackled by using one more laser. This way we have two signals so we can communicate four informations. In my course here, I'm studying a subject of Principals of programming language and there I studied a useful concept of Finite state automata. Although I had no idea of this when I originally developed the application, I have found that this concept can be used to give a theoretical background to the concept that I used.

When a person enters the room he will strike with one of the lasers and the MCU will change its state (to state *) and when the person strike the other laser the MCU (in state *) knows that the person has already struck with the other laser and so he's about to enter or leave. If the person is striking with the laser towards the room he's entering and if the person is striking with the other laser, he's leaving.



Components

When I did this project I had little experience with the components that are available in the market. So I used all the components of a 'Line follower robot' that I had from a workshop of robosapians in my college.



I had a controller kit that contains an ATmega8 MCU and L293D motor driver IC that was used to drive DC motors. I used this L293D IC to drive the relay interfaced to tubelight. The basic idea was that if the L293D IC can drive the DC motors, it was also expected to drive a relay as well and it did!!

Apart from that I had a pair of IR sensors that contains a TX and RX part moulded together so that it could be used as a proximity sensor or a line detector. I knew that it had a RX LED that was the photo diode so I had a intuition that it could also respond to a laser beam after changing the sensitivity level. This was actually the driving force of the project.



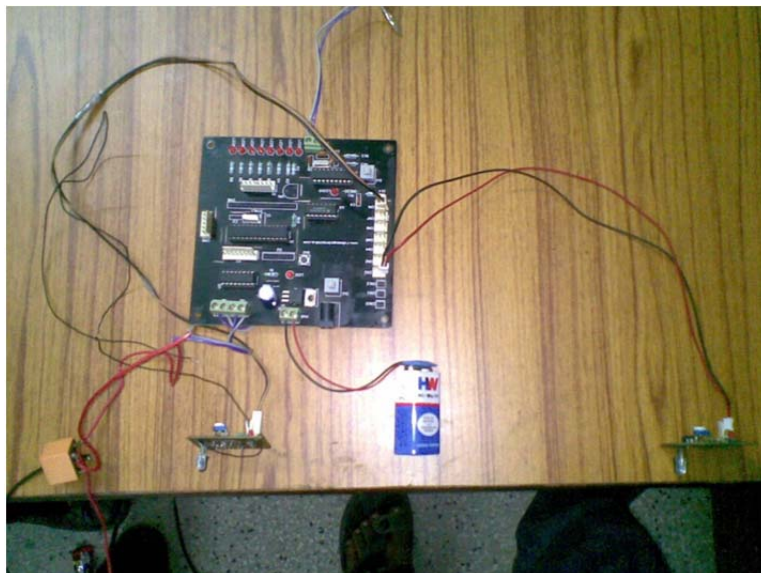
For that I brought a laser from the market and directed it on the RX and to my luck it was detected so I knew if I could interface a relay to the L293D it's all code that remains.

Arrangement

The arrangement was a bit of problem. Focusing the laser right into the IR sensors was very difficult. But still somehow I fixed both lasers to a table with the help of my friends who were really excited to see it working.



Having somehow aligned the lasers to the IR sensors on a similar table at the other end did the job.



Rest all was to open an electric box (I don't know what exactly it is known as) and cut a wire that goes to a switch and put the relay in parallel to the electric switch. This way I created another switch to the bulb socket. To be safe I used the neutral wire with the switch in OFF position and a socket on which no bulb was connected.

Code

I have been always very confident about my coding part. However in the end I always feel like it wasn't that simple as I expected in the start. However now I have gained enough experience and now I can better guess the difficulty level of the project.

Initially I thought I can do it using simple polling however due to poor response and accuracy and various time synchronous problems, I had to read all about interrupts and implement them decisively using properly calibrated delays at various points in the code.

I listed all the three version of successful code that I wrote during the development of the project, it shows how I worked it out finally. The final version of the project was very accurate and worked amazingly to keep track of number of people in the room and turn the light off when people becomes zero.

1st version

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>

void main()
{
    DDRD=0x00; //input port
    DDRB=0xff; //output port

    while(1)
    {
        roomside=PINC & 0b0001000;    //PC3
        doorside=PINC & 0b0010000;    //PC4

        if(roomside!=0)    //input from the sensor
        {
            sen_roomside=1;
        }

        if(doorside!=0)    //input from the sensor
        {
            sen_doorside=1;
        }

        if(sen_roomside==1 && sen_doorside==1)    //person is leaving
        {
            ppl--;
            sen_roomside=0;
            sen_doorside=0;
        }
        else
        {
            sen_doorside=1;    //set the flag of sensor1 wait for sensor 2
        }

        //drive the relay
        PORTD=0b00000001;    //LEDS
        PORTB=0b00000001;    //MOTOR
    }
    else
    {
        PORTD=0b00000000;
        PORTB=0b00000000;
    }
}
}
```

2nd version

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

char sen_roomside=0; //towards door room
char sen_doorside=0; //towards room room
int pp=2;

/*
INT0    PD2    pin4    PC4    27_PIN DOOR SIDE
INT1    PD3    pin5    PC3    26_PIN    ROOM SIDE
*/

ISR(INT0_vect) //door side ((sen1))
{
    if(sen_roomside==1 && sen_doorside==1) //person is leaving
    {
        PORTB=0b00000000;
        _delay_ms(2000);
        pp--;
        sen_roomside=0;
        sen_doorside=0;
    }
    else
    {
        sen_doorside=1; //set the flag of sensor1 wait for sensor 2
    }
}

ISR(INT1_vect) //room side ((sen2))
{
    if(sen_roomside==1 && sen_doorside==1) //person is entering
    {
        //PORTB=0b00000001;
        pp++;
        _delay_ms(2);
        sen_roomside=0;
        sen_doorside=0;
    }
    else
    {
        sen_roomside=1; //set the flag of sensor1 wait for sensor 2
    }
}

//1/5/10

void main()
{
    DDRD=0x00; //input port
    DDRB=0xff; //output port

    MCUCR |= (1<<ISC00) | (0<<ISC01);
    MCUCR |= (1<<ISC10) | (0<<ISC11);

    GICR |= (1<<INT0);
    GICR |= (1<<INT1);
    sei(); //SREG |= (1<<7);

    for(;;)
    {
        PORTB=0b00000000;
        _delay_ms(2000);
        if( pp<=1 )
        {
            PORTB=0b00000000;
            _delay_ms(9000);
        }
    }
}
```



```

        else
            PORTB=0b00000001;

    }

}

```

3rd version

```

#include <avr/io.h>
#include<avr/interrupt.h>

char sen_roomside=0; //towards door room
char sen_doorside=0; //towards room room
int pp=0;

/*
INT0      PD2      pin4      PC4      27_PIN DOOR SIDE
INT1      PD3      pin5      PC3      26_PIN   ROOM SIDE

*/

ISR(INT0_vect) //door side ((sen1))
{
    if(sen_roomside==1 && sen_doorside==1) //person is leaving
    {
        PORTB=0b00000000;
        // _delay_ms(2000);
        pp--;

        sen_roomside=0;
        sen_doorside=0;
    }
    else
    {
        sen_doorside=1; //set the flag of sensor1 wait for sensor 2
    }

    if( pp<=0 )
    {
        PORTB=0b00000000;
        //_delay_ms(9000);
    }

    else
        PORTB=0b00000001;
}

ISR(INT1_vect) //room side ((sen2))
{
    if(sen_roomside==1 && sen_doorside==1) //person is entering
    {
        //PORTB=0b00000001;
        pp++;
        //_delay_ms(2);
        sen_roomside=0;
        sen_doorside=0;
    }
    else
    {
        sen_roomside=1; //set the flag of sensor1 wait for sensor 2
    }

    if( pp<=0 )
    {
        PORTB=0b00000000;
        //_delay_ms(9000);
    }

    else
        PORTB=0b00000001;
}

void main()

```



```

{
    DDRA=0x00; //input port
    DDRB=0xff; //output port

    MCUCR |= (1<<ISC00) | (0<<ISC01);
    MCUCR |= (1<<ISC10) | (0<<ISC11);

    GICR |= (1<<INT0);
    GICR |= (1<<INT1);
    sei(); //SREG |= (1<<7);

    while(1)
    {
}
}

```

Conclusion

The application really appears 'cool' for what it was designed and very nicely counts the number of people in the room and turns the lights off when they reduce to zero. This project could be implemented at the main gate of the house so that when nobody is at home your smart home will know that it can turn various features ON like burglar alarm, turn OFF all lights and fans to save energy, etc.

However as I already mentioned my main aim of developing of developing the application was not for proper project implementation or commercial but to learn and enjoy the beauty of automation, I think I did it fairly good on that. As I thought and coded all the application myself it gave a great feeling at the end when I saw everything looking so cool.

Once in summer holidays I thought to implement again as a permanent solution in my house at the main gate and extent it to use it as a security system as well using a beautiful idea that I read somewhere on internet. For that I had to use another laser that focus on small mirrors and eventually creates a mesh of light. So someone entering the house when nobody is there will cause some alarm to trigger. However due to unavailability of components in my city and no soldering experience at that time, it remained an idea only.