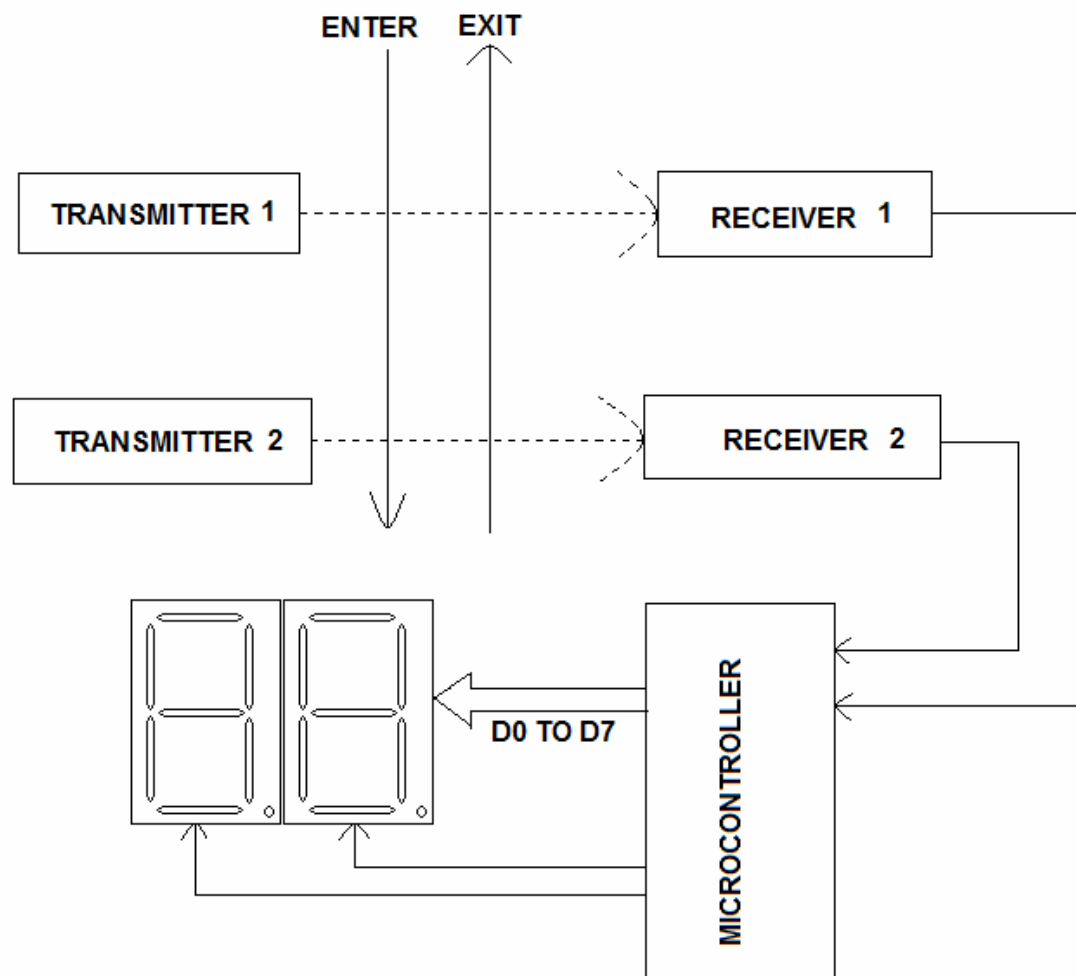


# **BI DIRECTIONAL VISITOR COUNTER**

## **1. INTRODUCTION**

This project is the most common and interesting to start with. The application is counting the number of persons entering in and exiting out like in Delhi Metro stations, Industries, offices, lift, car parking, and many more



**fig 1. BLOCK DIAGRAM**

Our objective is to count the objects (persons) entering and exiting the room so we need some sensors to detect the objects and a control unit which calculates the object, below you can find the block diagram and circuit diagram which illustrate the solution and the Embedded 'C' source code which calculate the object. Remember that this circuit is used with GP\_KIT\_MCS51-2.02 from BISD Labs, New Delhi; the kit contains rest

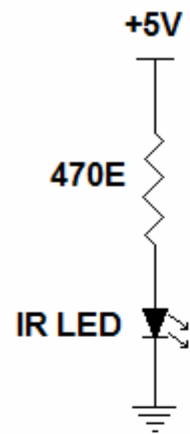
of circuit like 8051 microcontroller, interfacing two digit seven segments, on board voltage regulator to provide +5V D.C, ISP circuit, and a lot. Also refer the user manual of this kit.

### **PRINCEPLE:**

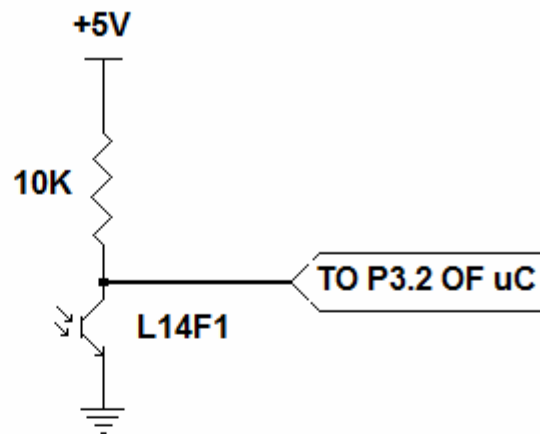
From the block diagram it is clear that the sensor pairs are placed face to face so that an IR radiations from IR LED are continuously received by phototransistor which makes its emitter base junction forward and collector current  $I_c$  equals to emitter current  $I_e$  (i.e,  $I_c=I_e$ ) assuming base current to be negligible. Hence the voltage at collector node becomes zero (logic 0) which is feed to microcontroller port pin P3.2 and P3.3, if any object is placed in between the sensor pair blocks the IR radiation which in turns put the phototransistor in cut-off mode and  $I_c \neq I_e$ , this makes collector voltage to +5V (logic 1)

In our program we have to poll both the inputs from both the sensors at port pin P3.2 and P3.3 to detect for the entry or exit, if sensor pair one is been obstructed (P3.2 becomes one) first, implies persons entry and second pair is obstructed (P3.3 becomes one) first shows exit. After obstructed any one sensor we have to poll for the next sensor to determine a complete entry or exit.

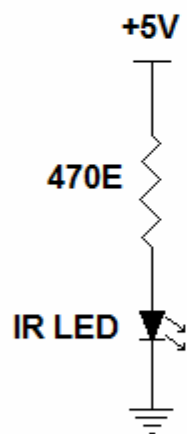
## **2. CIRCUIT IDEA**



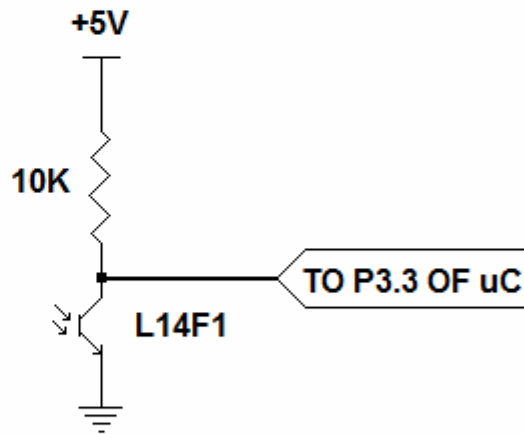
TRANSMITTER 1



RECEIVER 1



TRANSMITTER 2



RECEIVER 2



- 1. EMITTER
- 2. BASE
- 3. COLLECTOR

L14F1 BOTTOM VIEW

### 3. SOURCE CODE

```

#include<reg51.h>
sbit s1=P3^2;
sbit s2=P3^3;
sbit seg0=P1^0;
sbit seg1=P1^1;
unsigned char arr_seg[10]={0x24,0x77,0xa1,0x61,0x72,0x68,0x28,0x75,0x20,0x60};
char ch_l,ch_r,digit=0;
void isr_disp(void) interrupt 1
{
    if(digit==0)
    {
        seg0=0;
        seg1=1;
        P0=arr_seg[ch_l];
        digit++;
    }
    else if(digit==1)
    {
        seg0=1;
        seg1=0;
        P0=arr_seg[ch_r];
        digit=0;
    }
    TH0=0xdb;
    TL0=0xff;
}
main()
{
    unsigned char count=0;
    TMOD=0x01;
    TH0=0xdb;
    TL0=0xff;
    IE=0x82;
    TR0=1;
    ch_l=0;
    ch_r=0;
    while(1)
    {
        if(s1)
        {
            while(s1);
            while(!s2);
            while(s2);
            if(count<99)
                count++;
        }
        if(s2)
        {
            while(s2);
            while(!s1);
            while(s1);
            if(count>0)
                count--;
        }
        ch_r=count%10;
        count=count/10;
        ch_l=count%10;
    }
}

```