

VISITOR COUNTER USING MATLAB

■ KARTHIKEYAN N.

This circuit can also be used for object counting in conveyor systems, vehicle monitoring and intruder detection. It consists of a pair of IR transmitter and IR receiver – both built around timer IC NE555 (IC1 and IC2, respectively). The IR transmitter and receiver units are mounted on the opposite frames of the entrance door such that the light from IR LED1 transmitter falls

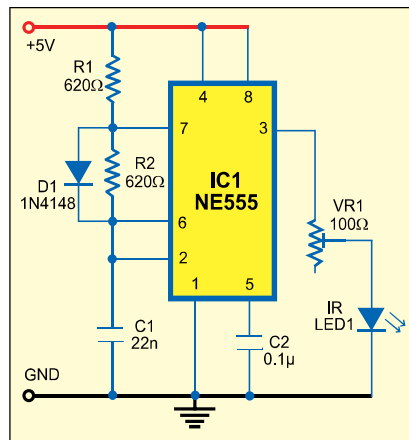


Fig. 1: Transmitter circuit

directly on the TSOP1736 receiver. The output of the receiver circuit is fed to MIC input of the PC. The program in MATLAB in the PC is used as the user-interface to count the number of visitors. The count can be stored in the PC, manipulated and used for statistical applications.

A visitor count is registered in the PC whenever someone blocks or interrupts the IR beam between the transmitter and the receiver. However, this does not indicate whether the visitor is going inside the room or leaving the room. This unidirectional counter counts the number of visitors within a range of about 1 metre between the transmitter and the receiver.

In the IR transmitter circuit (refer Fig. 1), IC NE555 is configured as an

astable multivibrator that oscillates at a frequency of 36 kHz. This frequency modulates the IR light beam emitted by IR LED1. The brightness of IR LED1 can be adjusted using variable resistor VR1.

TSOP1736 is an IR receiver normally used in TV remote control devices. It receives the IR signal transmitted by IR LED1. It is an active-low device, i.e., if it receives modulated IR light at a frequency of 36 kHz, its output pin 3 goes low. When it does not receive IR light at this frequency, its output pin 3 goes high.

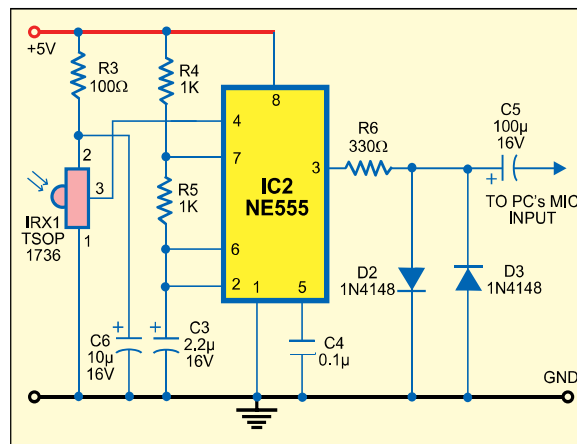


Fig. 2: Receiver circuit

Pin 3 of TSOP1736 is connected to pin 4 of IC2 (refer Fig. 2). Here too, NE555 is configured in astable mode. Whenever input pin 4 of IC2 goes high, an audible frequency of about 240 Hz is generated at its output pin 3. So when pin 3 of TSOP1736 is high, it enables IC2.

Output pin 3 of IC2 is connected to either MIC or line-in input of the PC's sound card. Diodes D2 and D3 protect the sound card from over-voltage.

Normally, the IR beam from IR LED1 is falling on TSOP1736 and therefore its output pin 3 is always low, disabling IC2. As a result, the PC doesn't receive the 240Hz sound signal.

When someone walks in to interrupt the IR beam, pin 3 of TSOP1736 goes high, which enables IC2 to gen-



erate 240Hz signal at its output pin 3. The PC receives the sound signal through the MIC input. The MATLAB program registers the signal input and increments the counter by one count.

The program increments the count only when it receives a sound signal followed by a 'no-signal' (silence) from the circuit. No-signal ensures that the person has crossed the door.

In the program, setting the lower and upper threshold values of the circuit power levels received by the PC is essential. The thresholds have to be set in the program by noting the power values during sound signal (upper threshold) and no-signal (lower threshold) at the MIC input. In this project, lower and upper threshold values are 4.05 and 172.0, respectively.

To check these values, the counter.m program code is used. Run this program in MATLAB version 7.0.1 and observe the values being displayed

PARTS LIST

Semiconductors:

IC1, IC2	- NE555 timer
IR LED1	- 5mm IR Light emitting diode
D1-D3	- 1N4148 signal diode
IRX1	- TSOP1736 infrared sensor

Resistors (all 1/4-watt, ±5% carbon)

R1, R2	- 620-ohm
R3	- 100-ohm
R4, R5	- 1-kilo-ohm
R6	- 330-ohm
VR1	- 100-ohm preset

Capacitors:

C1	- 22n ceramic
C2, C4	- 0.1μF, ceramic
C3	- 2.2μF, 16V electrolytic
C5	- 100μF, 16V electrolytic
C6	- 10μF, 16V electrolytic

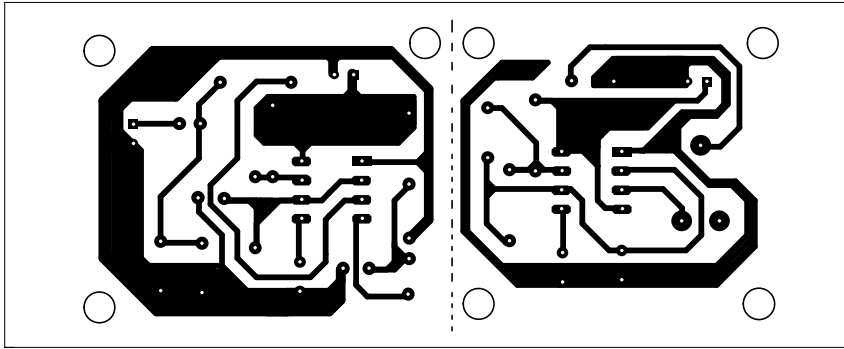


Fig. 3: An actual-size, single-side combined PCB for transmitter and receiver circuits

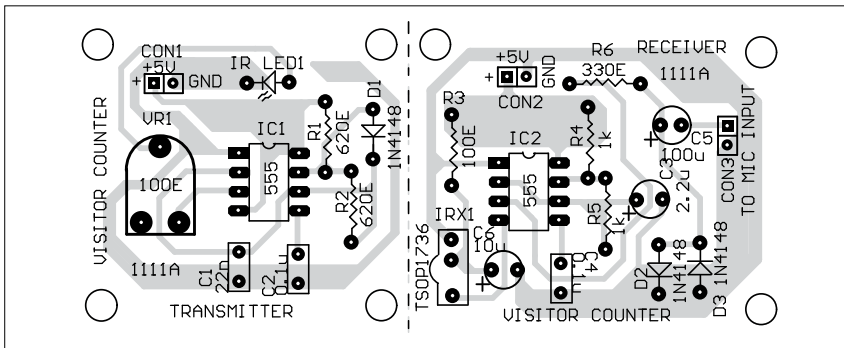


Fig. 4: Component layout for the PCB

on the screen. It generates the values in text format in command window. If the values obtained are not 4.05 and 172.0, check power supply to the transmitter

and receiver circuits (it should be 5V) and also vary VR1 in the transmitter circuit to get proper threshold values. Check whether there is any interfacing

problem with your sound card.

The threshold values have to be used in the main program (personcounter.m) as follows:

```
Fs=8000;
recordtime=1000;
lower=4.05;
upper=172;
PLAYSOUND=0;
```

Once you have done the settings as given above, run 'personcounter.m' and switch on the circuits to start counting. 'personcounter.m' is a GUI-based program that displays the number of visitors on the computer screen.

An actual-size, single-side combined PCB for transmitter and receiver circuits is shown in Fig. 3 and its component layout in Fig. 4.

The PCB for the transmitter circuit can be separated by cutting along the dotted line shown in Figs 3 and 4.

EFY note. 1. Don't forget to use diodes D2 and D3 in the circuit. Otherwise, your sound card may malfunction. It may even cause permanent damage to the sound card.

2. The source codes of this article are available on www.efymag.com website.

PERSONCOUNTER.M

```
function personcounter
Fs=8000;
recordtime=1000;
lower=4.05;
upper=172.00;
PLAYSOUND=0;
t= 1:2000;
sound = 2*sin(2*pi*t*1000/Fs);
count=0; %when you restart the program,
reload the old value here
sel=0;
STOPPED = 1;

% Figure Window
hfig = figure('Name','Person Counter',...
'Numbertitle','off',...
'Position',[600 500 350 100],...
'Menubar','none',...
'Resize','off',...
'KeyPressFcn',@keyPressFcn,...
'CloseRequestFcn',@closeRequestFcn);

% Menus
filemenu = uimenu('Label','File');
uimenu(filemenu,'Label','About','Callba
ck',@aboutFcn);
uimenu(filemenu,'Label','Exit','Callback',@
closeRequestFcn);

% Buttons
uicontrol(hfig,'Style','PushButton',...
'Position',[10 10 75 25],...
'String','START',...
'Callback',@startFcn);
uicontrol(hfig,'Style','PushButton',...
'Position',[140 10 75 25],...
'String','RESET (R)',...
'Callback',@resetFcn);
```

```
uicontrol(hfig,'Style','PushButton',...
'Position',[265 10 75 25],...
'String','EXIT (X)',...
'Callback',@closeRequestFcn);

% Stopwatch Time Display
DISPLAY = uicontrol(hfig,'Style','text',...
'Position',[10 45 330 55],...
'BackgroundColor',[0.8 0.8 0.8],...
'FontSize',35);

set(hfig,'HandleVisibility','off');

str = num2str(count);
set(DISPLAY,'String',str);

% Start the Timer
htimer = timer('TimerFcn',@
timerFcn,'Period', (recordtime+500)/Fs,'
ExecutionMode','FixedRate');
start(htimer);

function timerFcn(varargin)
if ~STOPPED
y=wavrecord(1000,8000);
mean(y(50:end,1));
power=sum(y.*y)
if(power<lower)
sel=1;
else if(power > upper &&
sel==1)
count=count+1;
sel=0;
str = num2str(count);
set(DISPLAY,'String',str);
if(PLAYSOUND)
wavplay(sound,Fs);
end;
end;
```

```
end;
end;
end;

function startFcn(varargin)
% Start the Stopwatch
STOPPED = 0;
end

function resetFcn(varargin)
% Reset the Stopwatch
STOPPED = 1;
count = 0;
str = num2str(count);
set(DISPLAY,'String',str);
end

function aboutFcn(varargin)
msgbox({' Person Counter Using
MATLAB',' GUI Version',',',...
' by Karthikeyan.N <karthi.
amrita.cbe@gmail.com>', 'About');
end

function closeRequestFcn(varargin)
% Stop the Timer
try
stop(htimer)
delete(htimer)
catch errmsg
rethrow(errmsg);
end
% Close the Figure Window
closeReq;
end
end
```