Week 4

Experiment # 10

Activity #1

Aim: Building and Testing the New IR Transmitter/Detector

Parts

S.No.	Componenets	Quantity
1	Piezoelectric speaker	1
2	Shrink wrapped IR LEDs	2
3	IR detectors	2
4	220 . resistors	2
5	Misc.Wires	required

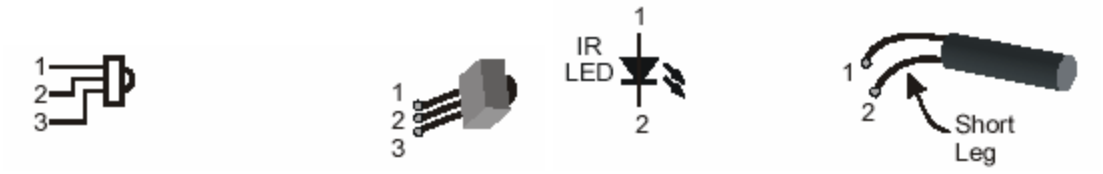

Figure1: IR detector schematic symbol and part on top row and IR LED schematic symbol and part on bottom row.

Circuit building: Object Detection Using Infrared

One IR pair (IR LED and detector) is mounted on each corner of the breadboard. Figure 2 shows the IR headlights circuit as both a schematic and wiring diagram. Build the circuit as shown.

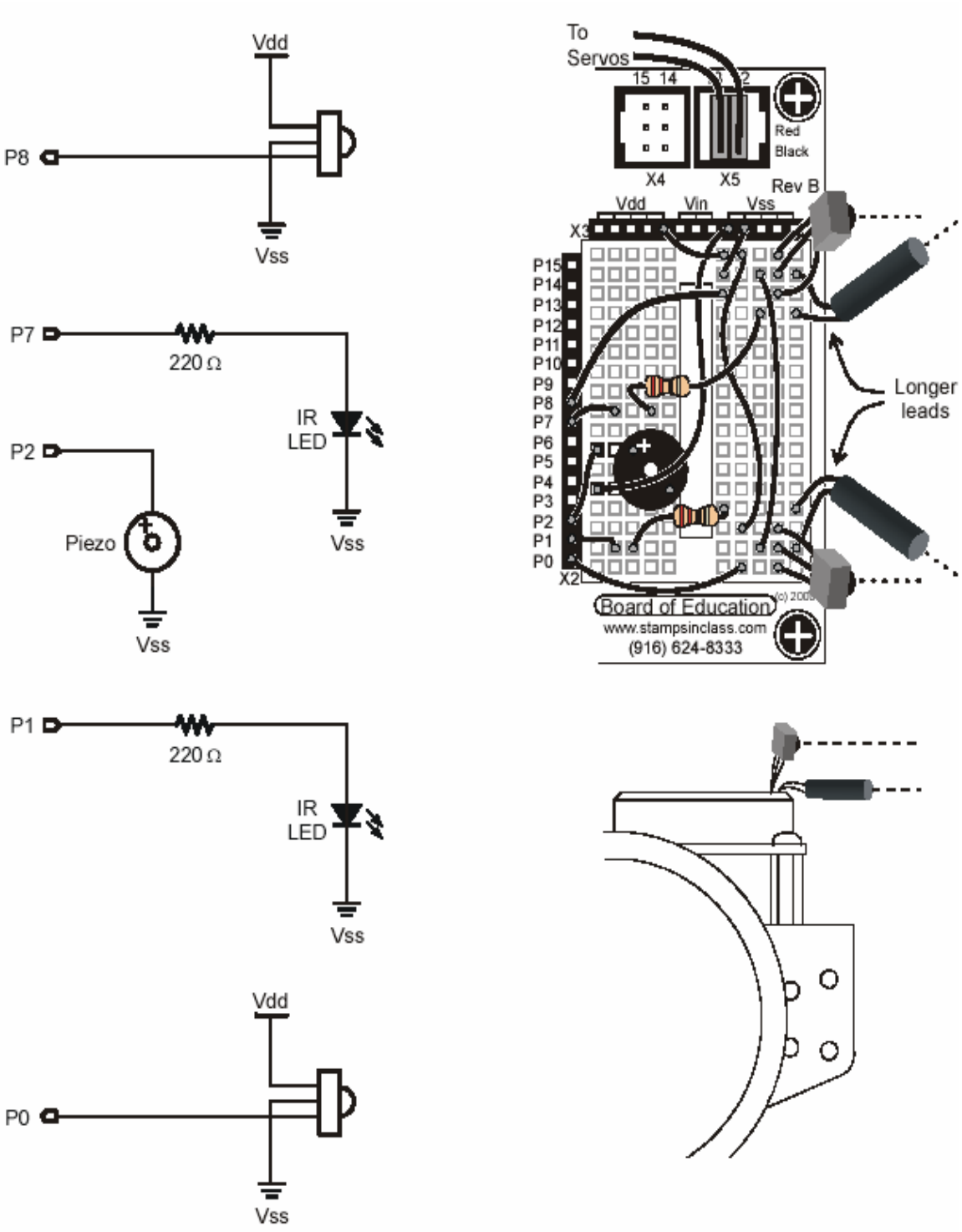

Figure 2: IR headlights (a) Schematic (b) wiring diagram.

Testing the IR Pairs

The key to making each R pair work is to send 1 ms of unfiltered 38.5 kHz **freqout** harmonic followed immediately by testing the signal sent by the IR detector and saving its output value. The IR detector's normal output state when it sees no IR signal is high. When the IR detector sees the 38500 Hz harmonic sent by the IR LED, it's output will drop from high to low. Of course, if the IR does not reflect off an object, the IR detector's output simply stays high. Program Listing 4.1 shows an example of this method of reading the detectors.

Enter and run Program Listing 3.1.

This program makes use of the Debug Terminal, so leave the serial cable connected to the BOE while Program Listing 3.1 is running.

```
' Robotics! v1.5, Program Listing 4.1: IR Pairs Display.

' {$Stamp bs2}                                    ' Stamp Directive.

'----- Declarations --------------

left_IR_det var bit                               ' Two bit variables for saving IR
right_IR_det var bit                              ' detector output values.

'----- Initialization ------------

output 2                                          ' Set all I/O lines sending freqout
output 7                                          ' signals to function as outputs
output 1

'----- Main Routine --------------

main:


                                                  ' Detect object on the left.
freqout 7, 1, 38500                               ' Send freqout signal - left IRLED.
left_IR_det = in8                                 ' Store IR detector output in RAM.
                                                  ' Detect object on the right.
freqout 1, 1, 38500                               ' Repeat for the right IR pair.
right_IR_det = in0

debug home, "Left= ", bin1 left_IR_det
pause 20

debug " Right= ", bin1 right_IR_det
pause 20
goto main
```

1. While program Listing 4.1 is running, point the IR detectors so nothing nearby could possibly reflect infrared back at the detectors. The best way to do this is to point the Boe-Bot up at the ceiling. The Debug output should display both left and right values as equal to "1."
2. By placing your hand in front of an IR pair, it should cause the Debug Terminal display for that detector to change from "1" to "0." Removing your hand should cause the output for that detector to return to a "1" state. This should work for each individual detector, and you also should be able to place your hand in front of both detectors and make both their outputs change from "1" to "0."
3. If the IR Pairs passed all these tests, you're ready to move on; otherwise, check your program and circuit for errors.

How the IR Pairs Display Program Works

Two bit variables are declared to store the value of each IR detector output. The first **freqout** command in the **main** routine is different. The command **freqout 7, 1, 38500** sends the on-off pattern shown in Figure 5.2 via left IR LED circuit by causing it to flash on and off rapidly. The harmonic contained in this signal either bounces off an object, or not. If it bounces off an object and is seen by the IR detector, the IR detector sends a low signal to IO pin P8. Otherwise, the IR detector sends a high signal to P8. So long as the next command after the **freqout** command is the one testing the state of the IR detector's output, it can be saved as a variable value in RAM. The statement **left_IR_det = in8** checks P8, and saves the value ("1" for high or "0" for low) in the **left_IR_det** bit variable. This process is repeated for the other IR pair, and the IR detector's output is saved in the **right_IR_det** variable. The **debug** command then displays the values in the debug window.

Task

Experiment with detuning your IR pairs by using frequencies above 38.5 kHz. For example, try 39.0, 39.5, 40.0, 40.5 and 41 kHz. Note the maximum distance that each will detect by bringing an object 0progressively closer to the IR pairs and noting what distance began to cause the IR detector output to switch from "1" to "0."