**ME 3200 Mechatronics I Laboratory**
# Lab 6: Introduction to Photosensors

**Introduction**

Autonomous robots are often programmed to be attracted to specific kinds of external sources: such as light, sound, temperature, elevation, etc.  In order to attract these robots to these sources designers use specialized sensors that are sensitive to the desired source.  In this laboratory exercise, you will learn how to use photosensors in conjunction with the Handy Board to detect the presence of infrared light.  In addition, you will explore the characteristics of the photosensors that will be used for your project and how you might use these sensors to drive your robot to a source of infrared light.  The techniques used in this lab to explore a sensor's capabilities are also applicable to other types of sensors—such as gas, sound, and temperature detection.

## Background

Photosensors come in a variety of types, but the one thing that they all have in common is that they are used to detect the presence of some kind of light.  Some are used to detect visible light; some detect ultraviolet light; the photosensors we use in lab, however, detect infrared light. These infrared sensors are sometimes referred to as infrared (IR) detectors.  The photosensors used in this lab are infrared detecting phototransistors as depicted in Figure 1 below.
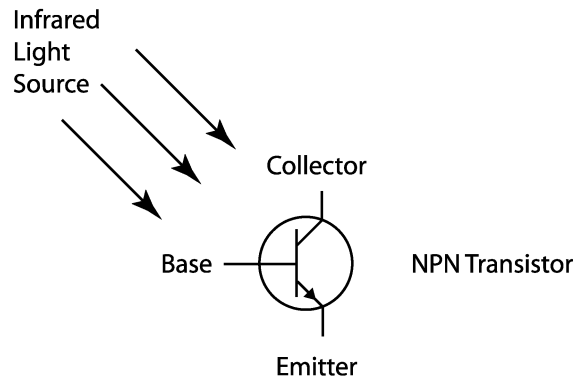


**Figure 1:  Schematic of an infrared detecting phototransistor.**

As the base of the phototransistor is exposed to infrared light, the base "closes" a switch between the collector and emitter. If a voltage difference is applied across the IR sensor, current is allowed to pass from the collector to the emitter.  The amount of current allowed to pass through the sensor is proportional to the amount of infrared light received by the base: more light produces a higher current.  Because of this phenomenon, the phototransistor can be viewed as a variable resistor.  When there is no infrared light present, the resistance is relatively high.  When the IR sensor is saturated with infrared light, the resistance is much lower.  This type of sensor is ideal for use with the Handy Board and its analog input ports.
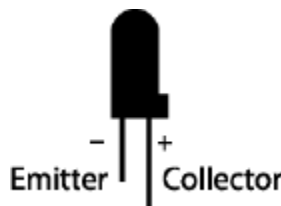


**Figure 2:  An infrared detecting phototransistor.**

## IR Sensors and the Handy Board

Before using these IR sensors with the Handy Board, the user must understand how to properly connect the IR sensor and the Handy Board and how they behave when connected. Start by examining the IR sensor and determining which of its two leads are respectively the emitter and the collector. Figure 2 depicts the difference between the two; remember that the collector lead is longer than the emitter, and that the emitter side of the sensor has a flat face. Once these leads are identified, the IR sensor is connected to one of the Handy Board analog input ports according to Figure 3 below.
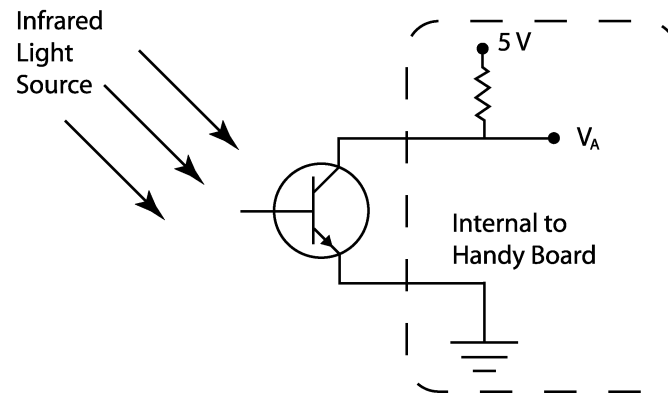


**Figure 3: The IR sensor and Handy Board combining to form a voltage divider.**

Once the sensor is in place, the internal voltage supply and pull-up resistor of the Handy Board combine with the sensor to form a voltage divider that determines the voltage, $V_A$, read by the Handy Board's eight-bit ADC (see Figure 3 above). To determine the infrared light level, use the `analog(p)` command where $p$ is an integer indicating which analog port to sample (refer to the number printed on the circuit board below the port). An integer will be returned that corresponds to the voltage, $V_A$, which is proportional to the effective resistance of the light sensor. In other words, in the absence of infrared light, the effective resistance of the sensor will be relatively high returning an integer near 255; when the sensor is saturated with infrared light, the integer returned will be very near zero. Recall that the analog-to-digital converter in the Handy Board will always return an integer value between zero and 255. It may not be very intuitive, but the integer returned from the `analog(p)` command is inversely proportional to the infrared level present.

Whenever IR sensors are used it is important to consider which sources of infrared light are present in the operating environment that may influence the response of the sensor. The best infrared sources are light sources that generate a lot of heat; incandescent light bulbs and the sun are the best examples. Florescent lights, however, produce very little infrared light. Whatever infrared light sources are present, it is always good to determine how they will affect the behavior of the IR sensors.

---

## Pre-Lab Exercise

Bring your completed rescue victims to lab with batteries installed ("completed" implies that you will have made the modifications to your victims since the last time your TA inspected them). Make sure that your victims are as professional looking as possible—holes sewn up, switches installed (if applicable), battery packs secure, etc.

---

## Laboratory Exercise

Equipment needed:
- Handy Board
- Telephone cord
- A 12 Volt, 500mA DC transformer
- Your Rescue Victims
- Infrared Sensors
- CutieBot
- Paper Compass (see the link just below the link to the Photosensor Lab on the Lab Handouts page of the class website.)

1) Interface your Handy Board with your computer using the same procedure from the *Introduction to the Handy Board* lab, place the Handy Board in interactive mode, and start Interactive C.

2) Insert the collector of an IR sensor into an analog input port, and insert the emitter into the ground bus.  Refer to Figure 2 above to help determine which lead is the anode and which is the cathode.

3) Use the `analog(p)` command to determine the ambient infrared light level in the lab.  Record the reading below:

4) Using the flashlight, saturate the IR sensor with the light and run the `analog(p)` command again.  Record the sensor reading below:

5) Write a program using the sample code in Figure 4, that averages five light readings of the infrared level of the center IR sensor of the CutieBot, normalizes the average (from 0 to 1), and prints the result to the LCD screen. Load the program into the Handy Board, demonstrate it to your TA.

   **Note:** The CutieBot uses five IR sensors distributed in increments of 22.5° along the front of the CutieBot.  The IR sensors from right to left are connected to analog ports 0 through 4 of the Handy Board of the CutieBot.

```
int i;                                 /* i is a counter */
int avg=0,avg2=0;                      /* average center ir value*/
float center;                          /* value of the center ir */
int num=5;                             /* the number of readings taken */

void main() {
  printf("Press start to read center ir");
  while(!start_button()) {}            /* wait until start is pressed */
  while(1) {                           /* run in a continous loop */
    center = 0.;                       /* initialize center */
    if(start_button()) {               /* take readings if start is pushed */
      for(i=1;i<num;i++) {             /* loop for running num times */
      avg += 255 - analog(2);          /* sums the value of the num runs */
      }
      avg /= i;                        /* divide by i to take the average */
      avg2 = (int)((float)avg/.255);   /* rounds answer to 3 sig. figs. */
      center = (float)avg2/1000.;      /* a float with range of 0 - 1 */
      printf("C=%f\n",center);             /* print the results */
    }
  }
}
```

**Figure 4: Sample code used to normalize the IR sensor output.**

6) Using a CutieBot, the program you wrote in the step five, your rescue victims, and the paper compass provided by the TA determine the IR detector's angular response characteristics.

a) Place the CutieBot on the floor on top of one of the paper compasses with the center IR directly over the 90° point and so that the wheel axles are parallel to the 0° - 180° line.

b) Using one of the lines in the floor as a guide set one of your rescue victims on the floor at least three feet directly in front of the CutieBot.

c) Using the paper compass, a line on the floor as a guide, and the program you wrote in step five, record the IR level at each mark on the compass. Record these data in the table below.

| Angle [degrees] | IR Level | | Angle [degrees] | IR Level |
|---|---|---|---|---|
| 0 | | | 95 | |
| 10 | | | 100 | |
| 20 | | | 105 | |
| 30 | | | 110 | |
| 40 | | | 115 | |
| 50 | | | 120 | |
| 60 | | | 130 | |
| 65 | | | 140 | |
| 70 | | | 150 | |
| 75 | | | 160 | |
| 80 | | | 170 | |
| 85 | | | 180 | |
| 90 | | | | |

7) Create a polar plot of the data collected above; Excel does not support a polar plot format so you will need to create the plot in Matlab.

a) To create this plot, use excel to make a file where column 1 lists the angle in radians and column 2 lists the normalized IR readings; save the file in .txt format to disk.

b) Open Matlab and set the current directory to the folder in which you saved your data file in the previous step.

c) Enter the following commands into the command window.

```
load yourfilename.txt
polar(yourfilename(:,1), yourfilename(:,2))
```

The plot will come up in a new window labeled Figure 1. Save the plot to your disk and compare your plot with Figure 2 of the IR sensor datasheet (a link to the datasheet is included on the Lab Handouts page of the class website).

8) Place one of your victims directly in front of the CutieBot at varying distances (between six inches and four feet), record the IR sensor readings from your program at each distance in the space provided, and comment on the reliability of the sensor to determine the distance to your victim.

| Distance | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| IR Reading | | | | | | | | | | | |

9) Write and implement an Interactive C program that will make a CutieBot search, find, and approach one of your rescue victims. Ideally, the CutieBot should stop when it is about a foot away from the victim. This task can be accomplished with many different control strategies: consider some of the following strategies—or create one of your own.

- When the program starts, the CutieBot scans the area around itself to find the victim initially. Then have the CutieBot approach the victim correcting its path until it reaches the victim using one of the strategies below.

- Compare the averages of the right and left sensors to the value of the middle sensor. Based on which region of sensors sees the most light, have the motors favor that side. The right motor is controlled by motor port 1, and the left motor is controlled by motor port 2.

- Have the CutieBot turn toward the sensor that has the highest IR sensor reading—turning right or left if the center sensor doesn't have the highest IR reading.

- Have the CutieBot stop when the center sensor is saturated or greater than a specified level as determined by your measurements in part eight.

Try to make the code as simple as possible to improve efficiency.

## Questions

1. Based on your experiments, how much variability in light sensor readings would you expect for repeated measurements of a constant light source? Are there any advantages/disadvantages of averaging multiple readings from the same IR sensor?

2. Based on your experiments, are the IR sensors good sensors to determine the distance to a light source in the environment? Why or why not?

3. Compare the angular resolution of the IR sensor as stated in the datasheet with your experimental results. How are they similar/different? Why?

4. Why does the Handy Board return a small number with high IR light levels and a high number with low IR light levels?

5. Explain the strategy that you found most successful for making the CutieBot find your victim.