ME 3200 Mechatronics I Laboratory *FALL 2002*

Lab 6: Introduction to Photosensors

In this laboratory exercise, you will 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.

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 light. Some are used to detect visible light, some detect ultraviolet light, but our photosensors detect infrared light. Some terms that are interchangeable with photosensors are infrared or IR detectors or sensors, and light sensors. The photosensors used in this lab are infrared detecting phototransistors as depicted in Figure 1 below.

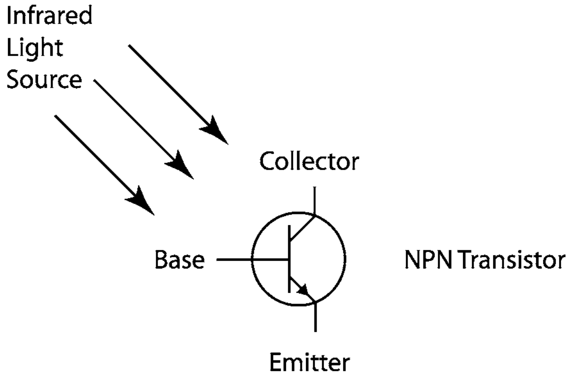

Figure 1. Schematic of an infrared detecting phototransistor.

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 detector, current is allowed across the phototransistor. 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 in the Handy Board analog input ports.

Laboratory Exercise:

Equipment needed: Handy Board, infrared light source, and photosensors.

1. Interface the Handy Board to the computer and place it in interactive mode as you did in the Introduction to Microcontrollers experiment.
2. Insert the cathode (collector) of an IR sensor into an analog input port, and insert the anode (emitter) of the sensor into the ground bus. Refer to Figure 2 to help determine which lead is the anode and which is the cathode. It is wise to remember that the cathode lead is longer than the anode, and that the anode side of the sensor has a flat face.

   Refer to Figure 3 to determine how the phototransistor is inserted into the Handy Board.

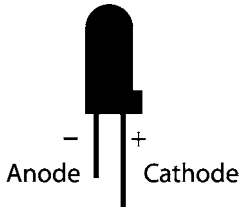

Figure 2. An infrared detecting phototransistor.

3. Once the sensor is in place, the internal voltage supply and pull-up resistor combine with the sensor to form a voltage divider to assist in reading the sensory input. The voltage, V_a, in Figure 3 below determines the infrared level the Handy Board will measure.

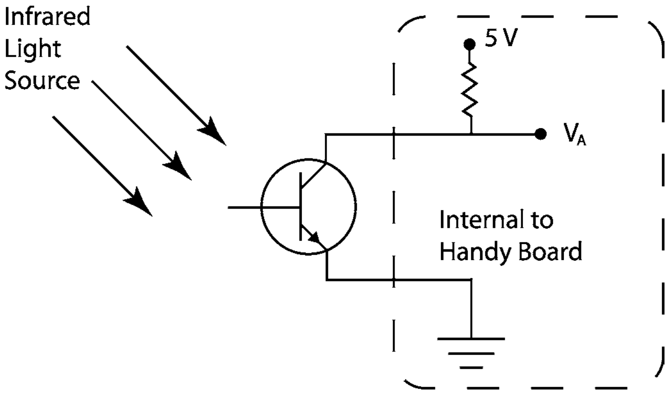

Figure 3. The photosensor and Handy Board combining to form a voltage divider.

4. To read the ambient infrared light level, use the command `analog(int p)` where *p* is the port used (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 a high integer; when the sensor is immersed in infrared light, the integer returned will be much lower. It may not be very intuitive, but the integer returned from the analog command is inversely proportional to the infrared level present. Recall that the analog-to-digital converter in the Handy Board is an 8-bit device, so the integer returned will always be between 0 and 255.

5. Ask the TA for a flashlight, shine it on the ir sensor and try the `analog(int p)` command again. Notice the difference it makes in the integer returned. The florescent light in the ceiling produces very little infrared light, but the incandescent bulb in the flashlight produces quite a bit of infrared light.

6. Write a program based on the sample code in Figure 4, that continuously reads the infrared level of the center IR sensor, normalizes it – from 0 to 1 – and outputs it to the LCD screen. Load the program into the Handy Board and run it.

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

void main() {
  printf("press start to  read center ir");
  while(!start_button()) {}     /* hold here 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 IR sensor code.

7. Use a CutieBot to explore the IR detector characteristics. A CutieBot uses five IR sensors to detect the presence of light. These sensors are 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.

   Using the paper compass and flashlight provided by the T.A., create an angular response curve for the center IR sensor on the CutieBot as in figure 2 of the IR datasheet that found on the lab web page. This can be done by placing the CutieBot on the floor on top of one of the paper compasses with the center IR directly over the point where the compass lines converge with the wheel axles parallel to the 0° - 180° line. Now set the flashlight on the floor at least ten feet from the CutieBot with the beam shining down one of the lines in the floor. Using the paper compass and the line on the floor as a guide record the IR level at each mark on the compass. Excel does not support a polar plot format so you will need to create the plot in Matlab. To create this plot, use excel to make a file where column 1 lists the angle in degrees and column 2 lists the normalized IR readings; save the file in .txt format. Next, open Matlab and set the current directory to the folder in which you saved your data, then 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. Compare your plot with figure 2. of the datasheet. Explain any discrepancies in the discussion section of your paper.

8. Write and implement a program that will get a CutieBot to chase a flashlight. You might try to compare the averages of the right and left sensors to the value of the middle sensor. Based on which region of sensors is experiencing the most light, have the motors favor that side. It is recommended that both motors run continuously. Just give one motor a higher signal level that the other side. In the case of the center sensor receiving the most light, drive both motors at the same rate. The left motor is controlled by motor port1, and the right motor is controlled by motor port 2.

Questions

1. Comment on the variability of a single light sensor reading. What are the advantages and disadvantages of taking an average of multiple sensor readings on your competition robot?

2. Are the IR sensors, as they are configured on the CutieBots, good sensors to determine the distance of the CutieBot from 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.

4. Why does the Handy Board return a small value with high light levels and a high value with minimal light?

5. Explain the strategy that you found most successful for making the CutieBot follow a light source.