Chapter 5

Sensor Design

Close your eyes. Plug your ears. Hold your nose. Tie your hands behind your back. Shut your mouth. Tie your shoelaces together. Spin yourself around a few times. Now walk. How does it feel? That's exactly what your robot feels: nothing–without sensors. You have been given many types of sensors that can be used in a variety of ways to give your robot information about the world around it. In this chapter, we'll explain each of the sensors you have, how it works, what it's good for, and how to build it.

Before we can teach you what sensors do, we need to make one point very clear. Sensors are not magical boxes. The phrase "Sensors indicate a large LEGO robot 2 meters off our port bow, Captain!" will never appear. All information you get from sensors must be decoded by *you*, the human builder and programmer.

Sensors convert information about the environment into a form that can be used by the computer. The sensors that are on the robot can be related to sensors found in humans. Touch sensors embedded in your skin, visual sensors in your retina, and hair cells in your ears convert information about the environment into neural code that your brain can understand. Your brain needs to understand the neural code before you can react. Since you will be programming the robot, you will need to understand the output of the sensors before you can program your robot to react to different stimuli.

Take time to play with each of the sensors you have been given. Figure out how they work. Look at the range of values they returns, and under what conditions it gives those values. The time you spend here will greatly ease your integration of hardware and software later. The better you understand your sensors, the easier it will be for you to write intelligible control software that will make your robot appear intelligent. So as you read about the sensors, you should assemble a bunch of sensors as shown.

Sensors provide feedback to your program about the environment. Feedback is important in any controlled situation. Rather than using open-loop, or timed programs that simply follow a pattern but have no real knowledge of the world, sensors can provide the feedback necessary to let a robot make decisions about how to act in its environment. The feedback mechanism is very important in an environment that is continually changing. During the rounds of the contest, the objects on the playing field will be changing their location (i.e., the other robot moves, the drawbridge closes, or you bump into a block). We strongly encourage you to use closed-loop feedback design when planning and implementing your strategy. There will be a smaller chance of random errors completely messing up your game if you use sensors wisely. See Chapter 8 for more information on the control problems you may encounter.

5.1 Sensor Assembly

You should have read the section on the previous chapter on the types of connectors used with the 6.270 board. This is an important concept to understand before building your sensors.

When building your sensors, **do not make your wires too long**. Excess wiring has a tendency to get caught in gears and other mechanisms. Start out with sensor wires no longer than 1 foot long and when your finally decide on your robot configuration, you can modify to length. Just build a few of each type so you can play with them.

Start out with building simple sensors like one or two switches. The more complicated ones will be the analog sensors that use IR.

5.2 Analog vs. Digital Sensors

The sensors you have can be split into two basic types: analog and digital. Analog sensors can be plugged into the analog sensor ports, which return values between 0 and 255. Digital sensors can be plugged into either the digital ports or the analog ports, but will always return either 0 or 1.

ANALOG $0 <= x <= 255$
DIGITAL 0 or 1

Each type of sensor has its own unique uses. Digital sensors, such as pushbuttons, can tell you when you've hit a wall. Digital sensors always answer a question about the environment with a *yes* or *no*. "Have I hit the wall?" Yes if the switch is closed, or no if the switch is still open.

Analog sensors, such as photoresistors, can tell you how far the sensor has bent, or how much light is hitting the sensor. They answer questions with more detail. Analog sensors, however can be converted to digital sensors using thresholding. Instead of asking the question *How much is the sensor bent?* you can ask the question: *Is the sensor bent more than half way?* The threshold can be determined by playing around with the specific sensor.

5.3 Location of Digital and Analog Ports

The digital ports on the main board are labeled from 0-7 and are shown in figure 4.13. There are also four analog ports on the main board, but when you use the expansion board, the analog ports get remapped to the connectors on the right side of the expansion board. The ports are all arranged in the same format. The inner most row of pins are the signals, followed by a space, then microprocessor power, and finally on the outer side is the ground.

5.4 Digital Sensors

Digital inputs all have *pull-up* resistors connected to them as shown in figure 5.1. Digital switches are wired such that the sensor is wired across the signal pin and ground. This means that when the digital sensors is closed, the signal is grounded or LOW. When the switch is open, the signal pin outputs +5V, or HIGH. This value is INVERTED by software, so reading the digital port with the switch open returns 0, while reading the digital port with the switch closed returns 1. With nothing plugged in, the value of a digital port should be 0.

Digital sensors can be used in the analog ports on the 6.270 Controller board as well, relieving any restrictions the small number of digital inputs may cause. Typical analog values for digital sensors are somewhat above 250 for an open switch, and less than 20 for a closed switch. When using the IC command, `digital(port)`–where port is an analog port number (i.e., greater than 7)–the sensor value is compared to a threshold value, and the command returns a 0 if the analog value is above the threshold or a 1 if the analog value is below it (remember the inversion of the actual signal that digital does?). This threshold's default value is 127, but it can be changed (See the section on IC commands for information on this).

A good way to get digital information from an analog sensor is to plug the analog sensor into a analog port and call it with the `digital(port)` command. For example, a reflectance sensor would return a 0 for black or a 1 for white if read with the digital command–provided the threshold is properly set. This can reduce some of the programming complexity by abstracting away the thresholding. You should however experiment with the sensors to determine the range of thresholds you get and under what conditions these thresholds are valid.

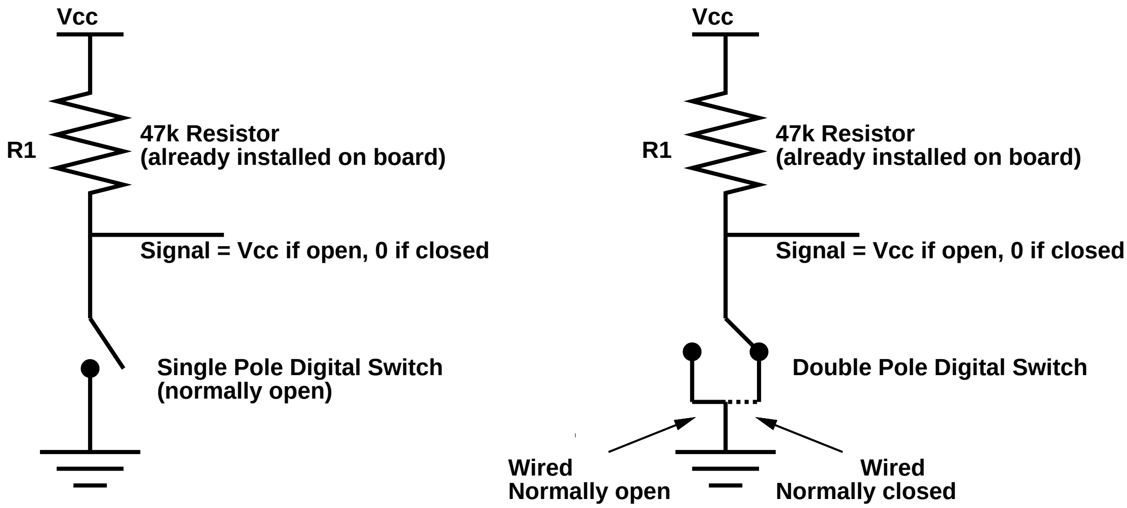

Figure 5.1: Generic Digital Sensor Schematics.

It is not recommended to plug analog sensors into digital ports, however, because the digital ports threshold to conventional logic levels which cannot be adjusted to suit each analog sensor. The valid analog readings may fall into the invalid range for digital logic.

Here are some mountings and uses for some digital sensors in the 6.270 kit.

5.4.1 Dip Switches

There are four dip switches on the Expansion Board. They can be used to select user program options during testing. One dip switch will be used in the starting code for the contest to determine the side your robot starts on and at which frequencies it transmits and receives the modulated IR. They can also be useful for outside control of program parameters, like enabling certain functions or selecting programs to run. While these switches are connected to the analog port, they are really digital switches.

5.4.2 Micro-Switches

The standard kit includes three types of small switches, two micro switches and a small push button. These make great object detectors, so long as you are only interested in answering the question, *Am I touching something right now?* with a *yes* or *no*. This is often enough for responding to contact with a wall or the other robot or for actuator position sensing. Using a switch in this manner (called a "limit" switch) can be a good way to protect drive mechanisms which self destruct when over driven. This could be handy for limiting the motion of hinged joints or linear actuators by

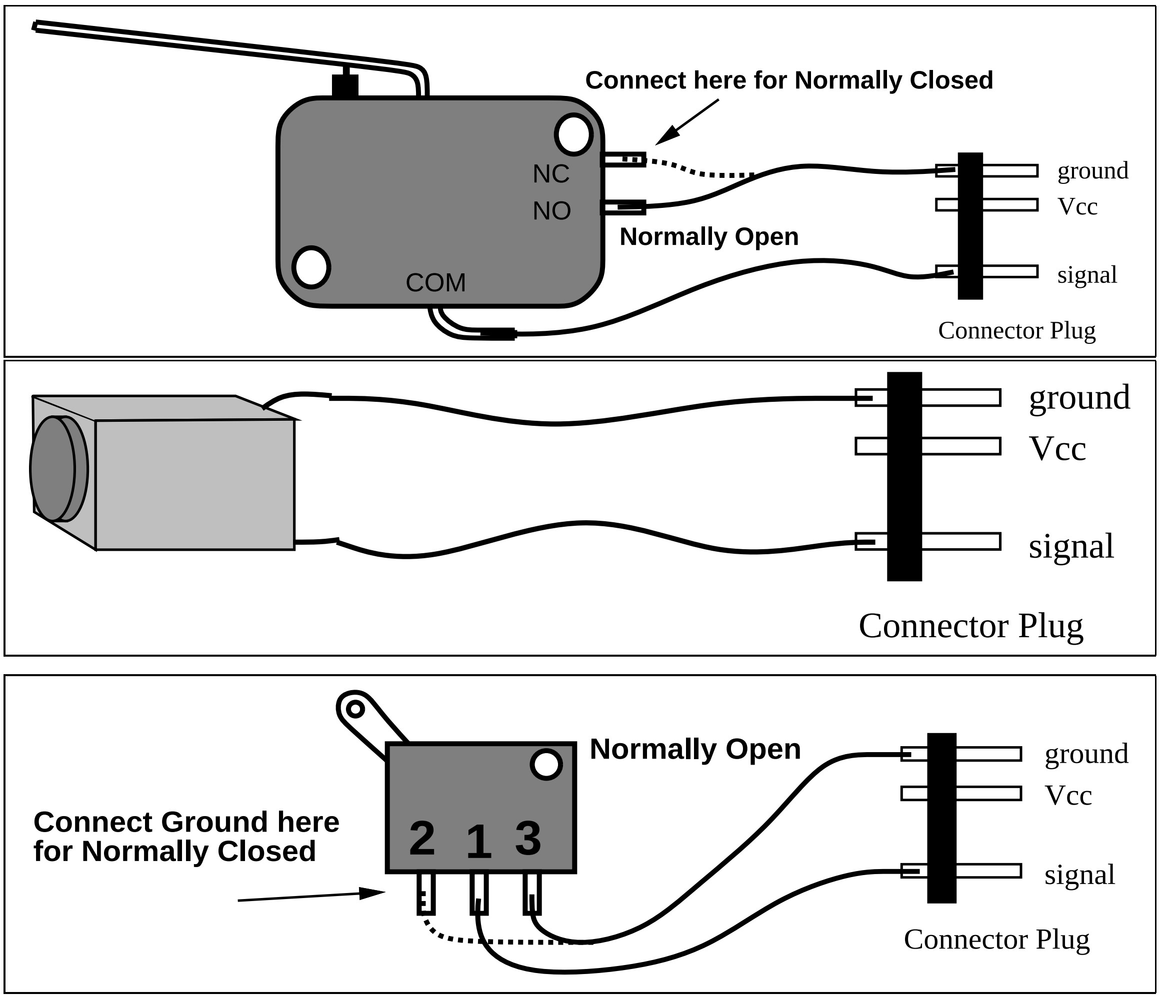

Figure 5.2: Microswitch Assemblies

requiring that a switch be open (or closed, depending upon the situation) before running the motor and monitoring it while things are moving. They could also be used for extended user interface for testing and development purposes.

The two micro switches are double pull, which means they can be wired so that they return a one or a zero when not depressed. The only major difference is how you think about the device in your code. Reading a sensor can be thought of as asking a question. Here, the question could be, "Are you open?" or "Are you closed?" If you wire the switch normally open, the answers are yes and no, respectively, where they would be no and yes for a switch wired normally closed, all for the same situation where the switch is not depressed.

Touch switches should be wired in a normally open configuration, so that the signal line is brought to ground only when the switch is depressed.

In some cases, a slight advantage may result from one arrangement, because there may be a difference between the position where the open side makes contact and the closed side breaks contact. When this is the case, the choice of normally open or normally closed will affect how sensitive the switch is to outside forces. This can allow you to make a very touchy sensing device or help block out noise. The small black switches with the white lever arm respond to a shorter arm movement when wired normally open and require a little more movement to cause a transition in the normally closed configuration.

Bouncing is a problem found in many switches. At the point where the switch goes from open to close or vice versa, the output from the switch is very glitchy. The switch may output several transitions. Bounciness occurs especially when the switch is used in a sensitive mode. One way to *debounce* the switch is to add a delay between samples of the digital input. If the sampling is sparse enough, the bouncing section of the data will not be collected.

5.4.3 Sharp IR Detector

The Sharp GP1U52X sensor detects infrared light that is modulated (i.e., blinking on and off) at 40,000 Hz. It has an active low digital output, meaning that when it detects the infrared light, its output is zero volts.

The metal case of the sensor must be wired to circuit ground, as indicated in the diagram. This makes the metal case act as a Faraday cage, protecting the sensor from electromagnetic noise.

While it may not seem like a digital sensor because most of the light sensor we deal with are analog, it is a bona fide digital sensor because it detects infrared light modulated at 40kHz. Inside the tin can, there is a IR detector, amplifier, and a demodulator. The sensor returns a HIGH when there is no 40kHz light, and is LOW when it see the 40kHz light.

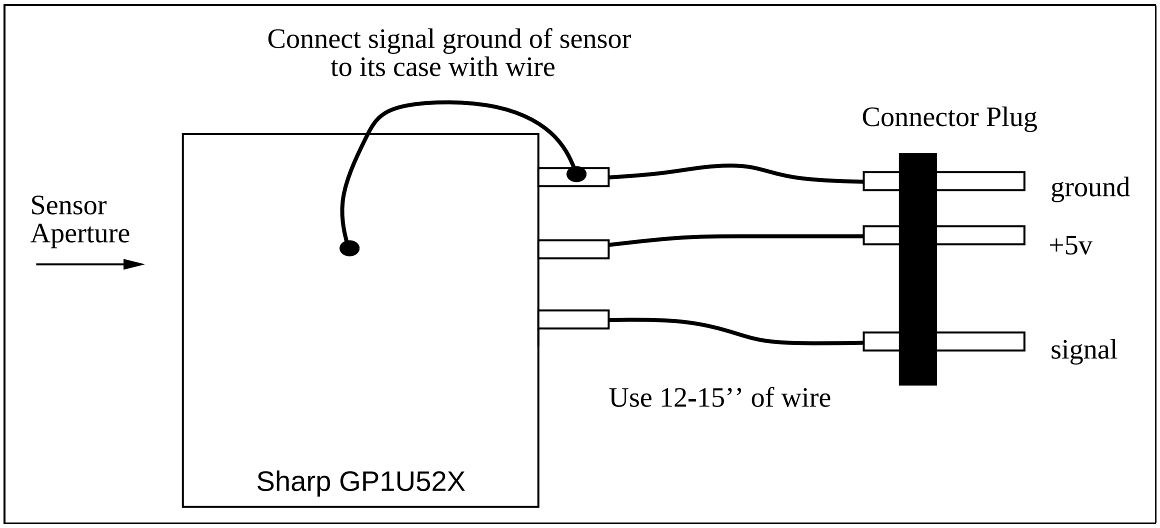

Figure 5.3: Sharp IR sensor assembly

There is a lot of infrared light that is ambient in the air. Some components of this light are at 40kHz, and straight output from the sensor would look very glitchy. The sun produces a lot of IR light, and in the sun, the sensor output bounces all over the place. To eliminate the effect of the stray IR light, the IR emitters are modulated at 100 or 125 Hz (see section A.7 for more information on the IR transmission) and the output of the IR Detectors is demodulated to look for these frequencies. The 40kHz frequency is known as the carrier frequency, and the other frequency is the modulated frequency.

You can use the IC command `ir_counts(port)` to count the number of successive detected periods of the modulated frequency. A count larger than 10 indicates a detection. You may need to play around with what values of the counts are needed for detection. These sensors can only be used in digital ports 4-7.

5.5 Analog Sensors

The analog ports all have a *pull up resistor* which is a 47KΩ resistor between +5 volts and the signal input. The analog readings are generated by measuring the amount of current flow through the pull up resistor. If no current flows through the resistor, the voltage at the signal input will be +5 volts and the analog value will be 255. The voltage at the signal pin can be simply calculated by:

$$V_{sig} = 5 - 47\Omega \times i$$

Reading the value of an analog port **without** a sensor will return a value above 250. With the sensor plugged in, the value should be less. This is one good way to

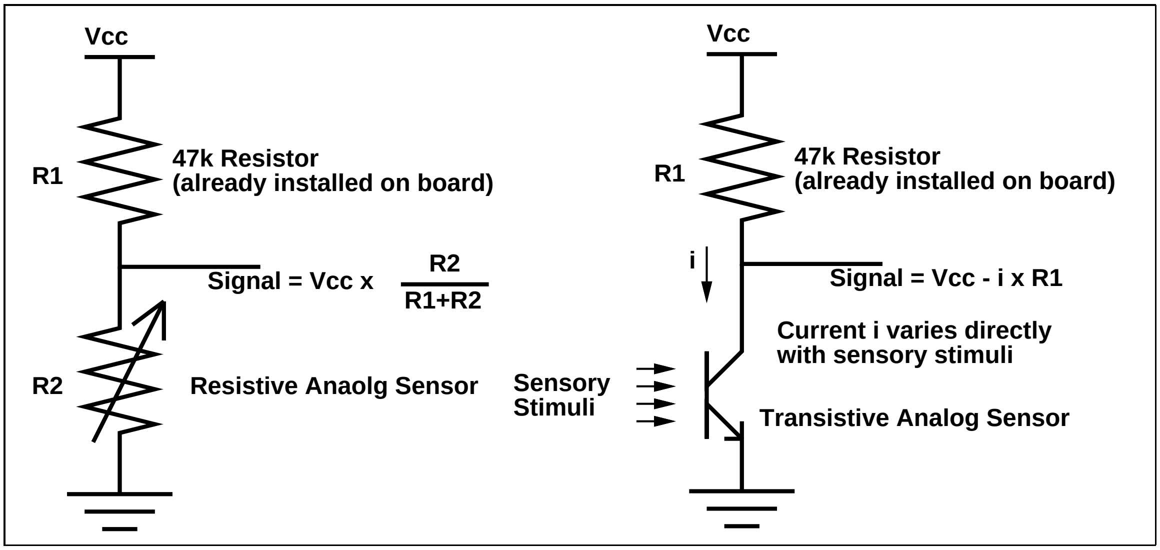

Figure 5.4: Analog Sensors Schematics

check if one sensor fell out: write a piece of code that checks the values of the analog ports that you have sensors plugged into. If that value is above 250 or so, have it tell you to check the sensor.

5.5.1 Resistive Sensors

The resistance of resistive analog sensors, like the bend sensors or potentiometers, change with changes in the environment, either an increase in light, or a physical deformation. The change in resistance causes a change in the voltage at the signal input by the voltage divider relation.

$$V_{sig} = \frac{R_{sensor}}{47\Omega + R_{sensor}} \times 5V$$

5.5.2 Transistive Analog Sensor

Transitive analog sensors, like the photo transistors and reflectance sensors, work like a water faucet. Providing more of what the sensor is looking for opens the setting of the valve, allowing more current to flow. This makes the voltage the voltage at the signal decrease. A photo transistor reads around 10 in bright light and 240 in the dark.

One problem that may occur with transitive sensors is that the voltage drop across the resistor may not be large enough when the transistor is open. Some transitive devices only allow a small amount of current to flow through the transistor. A larger range for the sensor can be accomplished by putting a larger pull-up resistor. By having a larger resistor, the voltage drop across the pull-up resistor will be proportional to the resistance.

We will give example uses and mountings for each type of sensor. Keep in mind that these are only simple examples and are not the only possible uses for them. It's up to you to make creative use of the sensors you've been given.

5.5.3 Potentiometers

Kits contain several sizes of potentiometers, also known as pots or variable resistors. There are rotary and linear pots. As the knob is turned or the handle slid, the resistance increases or decreases. This will produce different analog values.

Potentiometers should be wired with Vcc and ground on the two outside pins, and the signal wire on the center tap. This will, in effect, place the resistance of the potentiometer in parallel with the 47KΩpull-up on the expansion board and is more stable than just using one side and the center tab to make a plain variable resistor.

Potentiometers have a variety of uses. In the past, they have been used for menuing programs and angle measurement for various rotating limbs or scanning beacons. They can be used with a motor to mimic servos, but that's a difficult task. It is important to notice that the pots are not designed to turn more than about 270 degrees. Forcing them farther is likely to break them.

A potentiometer can be attached to a LEGO beam such that it can be used in place of a bend sensor. The rotation of the beam will produce a rotation in the potentiometer. See if you can come up with an assembly that can be used in place of a bend sensor. The advantage to such a sensor is that it is much sturdier than the bend sensor. The disadvantage is that it is bulkier.

Linear Pots

A linear potentiometer can be used to measure precise linear motion, such as a gate closing, or a cocking mechanism for firing balls or blocks.

Frob-knob

The frob knob is the small white dial on the lower left corner of the Expansion Board. It returns values between 0 and 255 and provides a handy user input for adjusting parameters on the fly or for menuing routines to select different programs. You may find it useful to glue a small LEGO piece to the frob knob to make turning it easier.

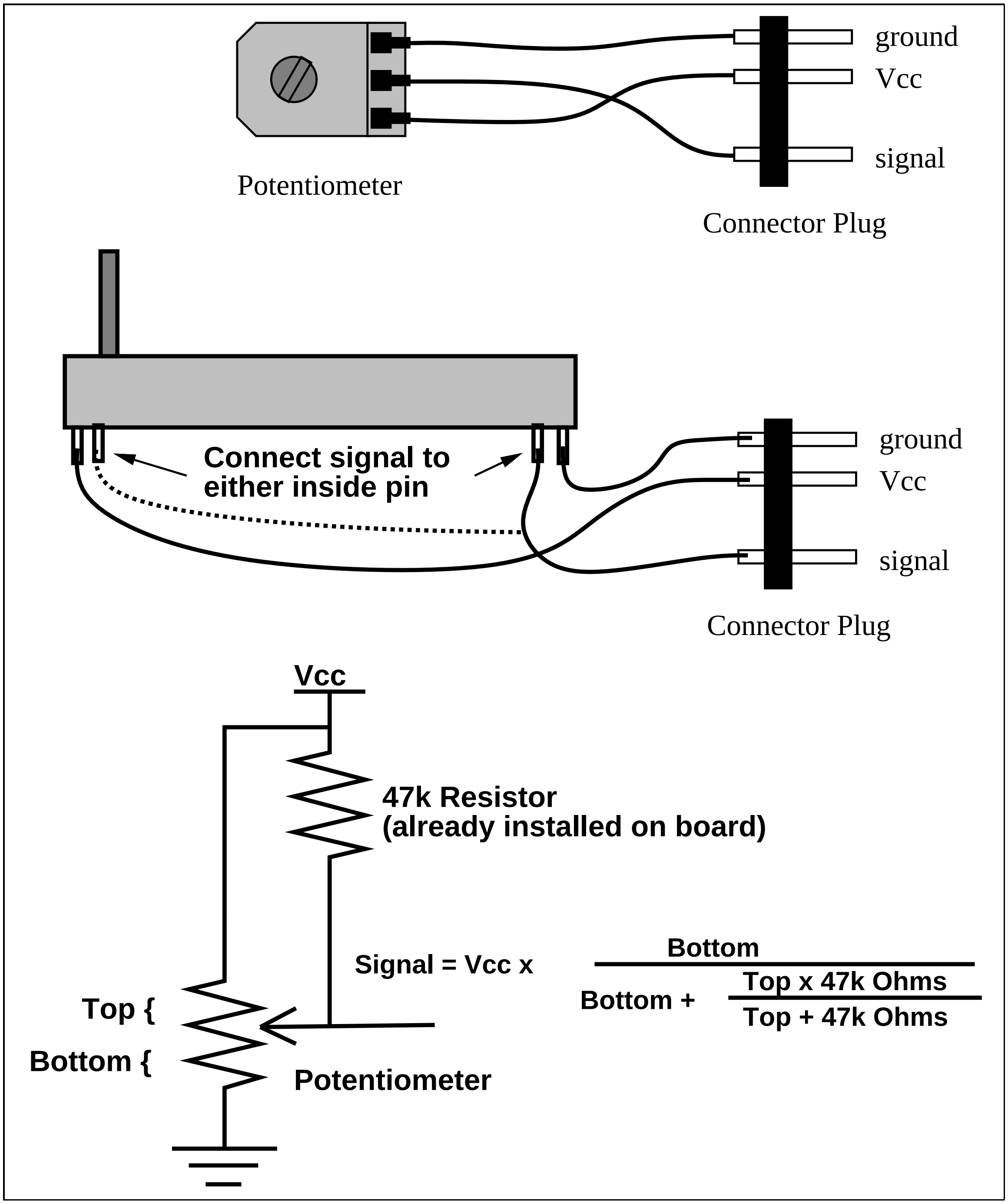

Figure 5.5: Potentiometer Assemblies

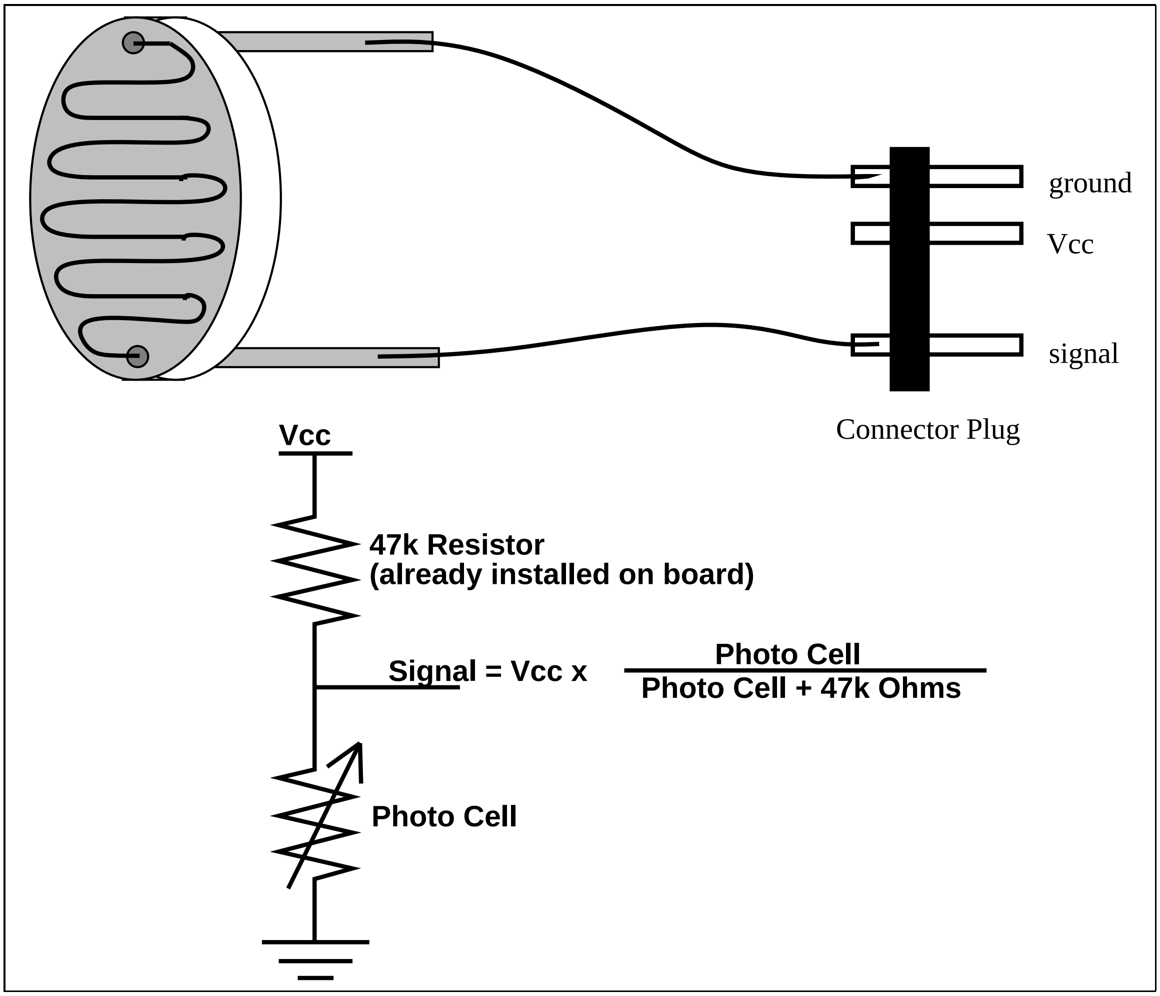

Figure 5.6: Photocell Light Sensor

5.5.4 Photoresistors

The photocell is a special type of resistor which responds to light. The more light hitting the photocell, the lower the resistance it has. The output signal of the photocell is an analog voltage corresponding to the amount of light hitting the cell. Higher values correspond to less light.

A photoresistor changes its resistive value based on the amount of light that strikes it. As the light hitting it increases, the resistive value decreases. They are somewhat sensitive to heat, but stand up to abuse well. Try not to overheat when soldering wires to them.

As with all the light-sensing devices, *shielding* is *very* important. A properly shielded sensor can make the difference between valid and invalid values reported by that sensor. The idea is simple: restrict the amount of light striking the sensor to the direction you expect the light to be coming from. You do not want light from external sources (i.e., camera flashes or spot lights) to interfere with your robot. Black heat shrink tubing often works well to shield the photoresistor from external light sources.

One good way to get a feel for how these sensors work, and how your robot and software interact, is to make a light-sniffer. With two or more photoresistors, try to create a simple robot that can move around a room, either avoiding light or avoiding shadows in a controlled manner. Ambient light conditions play a major role in how to interpret the data from any light sensors. A combination of photocells, one pointed up and one pointed down, may be used to adjust for ambient light levels, which may be useful in some applications.

Photoresistors are probably the only sensor required to be on your robot. A starting light will be used to start each contest round, and the robot must be able to sense that light. You must place one photoresistor on the underside of your robot, probably near the center. Be sure to shield it as much as possible from the overhead ambient light. We will provide starting code that reads the value of that sensor to start the match.

Mounting the photoresistors doesn't tend to be difficult. You can use a small amount of hot glue to attach the photocell to a LEGO brick, or double-sticky tape will also work. Be inventive.

5.5.5 Photo Transistors and IR LEDs

Phototransistors are usually tuned to a specific wavelength of light. The wavelength is usually near visible red, or in the infrared spectrum. They have similar properties to the photoresistors. The main difference is that the phototransistors are usually tuned to a specific wavelength. The other important difference is that the time delay for a change in light conditions is much smaller for a phototransistor. This can be useful in doing fast control looking for polarized light. The time constant for a phototransistor is much faster than a photoresistor, so it may be used in situations where timing is critical.

An IR LED is a type of diode which emits radiation in the infrared range. This part could be used as a component in a breakbeam sensor or a reflectance sensor.

These instructions are for two kinds of phototransistors, each of which are packaged in cylindrical brass-colored cans with a glass lens. The first kind is packaged individually, with no wires attached, and with three leads. The second is surplus parts, with wires already attached, and with each phototransistor paired with an LED. (Note: surplus parts are usually overstocked or obsolete parts that didn't sell through retail channels. See the book's appendix on ordering electronics parts.) The individual Phototransistors cost 6.270 about $1 each, about the same as an entire surplus assembly bundle of wires and phototransistors and LEDs.

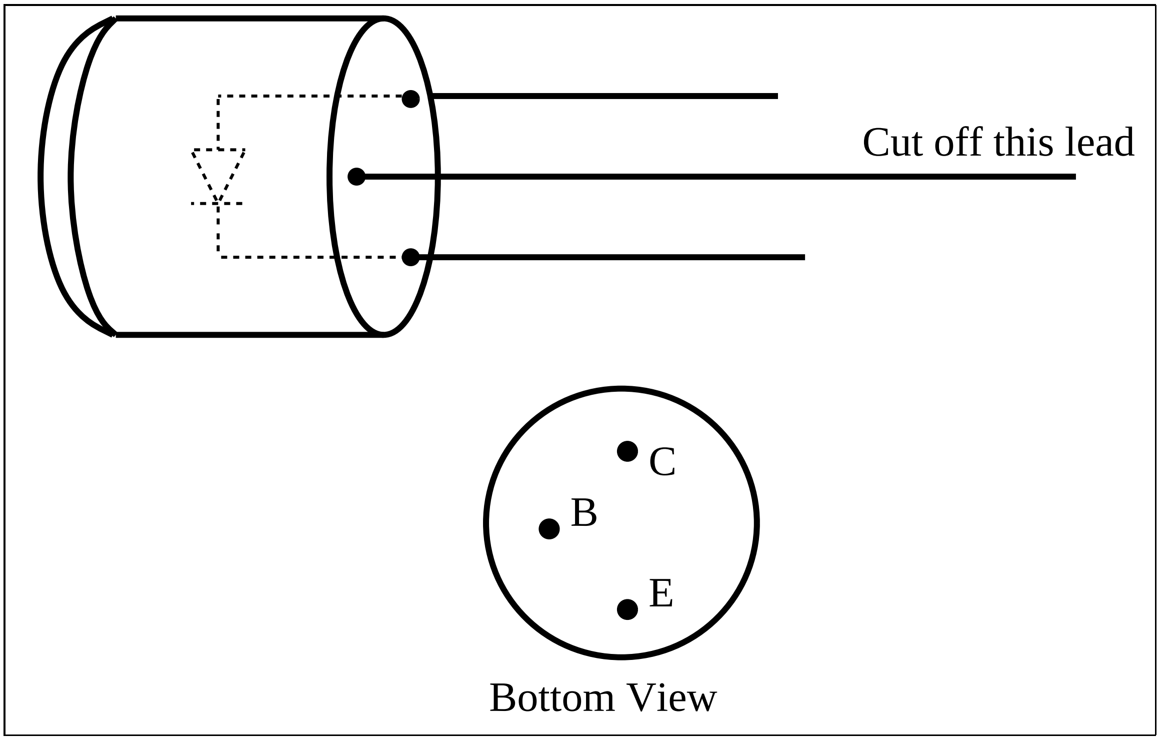

Figure 5.7: The LEDs

How to tell them apart

Be careful to differentiate the phototransistors from the LEDs: the phototransistors have relatively flat lenses, while the LEDS lenses are more convex. Fig 5.7 shows one of the LEDs. Also, the two different kinds of phototransistors (surplus vs virgin manufactured) have very different characteristics, and cannot be used in sensors interchangeably. The surplus phototransistors respond almost exclusively to infrared light and have a "resistance" of approximately 100 kΩ when activated and 1 MΩ when not activated. The individual, un-wired phototransistors, on the other hand, respond to visible light as well as infrared, and have "resistances" about one hundred times smaller.

Interfacing to the Board

These phototransistors require pull-up resistors, a resistor connected between Vcc and the signal line, to work properly. In past years, all of these sensors required 47k pull-up resistors, but that is no longer the case. Each individually packaged phototransistor now can be used with a 220k pull-up, while the "bundle of wires" phototransistors work well with 100Ωpull-ups. This may present A slight problem if you have already installed **RP6**, one of the 47k pull-up resistor packs on your expansion boards. Fear not! By installing the pull-up resistors on the connector as shown in Fig 5.8. For the individually packaged phototransistors, the 2.2k resistor on the connector will be in parallel with the 47k pull-up on the board. Since resistors in parallel add reciprocally, the combination of the two will electrically look like a 2.2k resistor (approximately). However, if you have the "bundle of wires" phototransistors, you will have to cut a trace on the bottom side of the expansion board to disable the 47k pull-up resistor, since it would otherwise dominate. Warning! Once you cut a trace, that analog port should be used only for the 220k phototransistors. This means that you will have to be sure to plug these sensors into the correct analog ports each time you use them. Ask a TA before you cut this trace!

Visible Light sensor

The phototransistors respond very well to visible (far-red, we hypothesize) light as well as infrared. They should be wired with a 2k to 4k resistor for best results (we recommend 2.2k). Because they respond to visible light, they are extremely susceptible to interference from ambient light. You may be able to use them as floor-color sensors using just ambient light, but if you want to use them for break-beam sensing, they will have to be very well-shielded.

IR Photo Transistor

The "bundle-of-wires" phototransistors are much more predictable. They should be wired with a resistor of 100k to 300k (we recommend 220k). They barely respond at all to visible frequencies of light. They respond particularly well to the LEDs with which they are bundled, as well as to the grey IR LEDs. Both LEDs are highly directional, and you should be able to get good break-beam results up to 5 or 6 cm apart (2 inches). This might prove especially useful in ball-firing mechanisms, for example. Note that both LEDs and phototransistors are just the right size to fit in LEGO axle-holes!

5.5.6 Polarizing Film

Polarized film has fine printed or etched straight lines. The polarizing film allows the light to travel in parallel perpendicular planes rather than in all directions. Assume for this section that the lines are running up and down, and therefore the light waves will be traveling up and down. If a second film is placed such that the lines are horizontal, the light traveling past the first filter will not pass through the second filter.

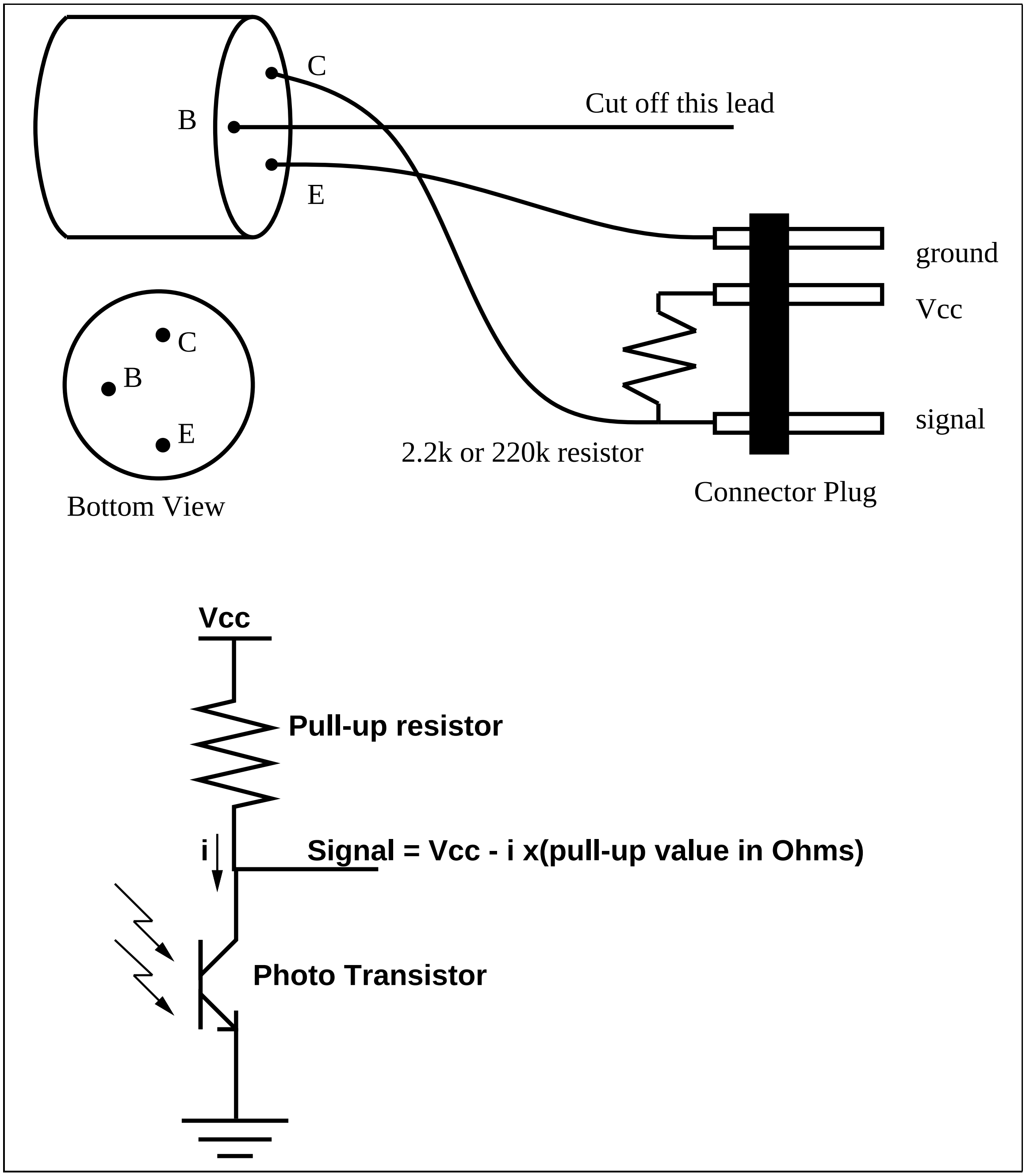

Figure 5.8: Phototransistor body and connector

Two pieces of film which are perpendicular to each other will block out most of the light. Parallel pieces will allow maximum light to go through.

The Polarizing film can be used to enhance the photo transistors and photo resistors. The beacons at each end of the playing field are emitting polarized light. One side is polarized at positive 45 degrees from the vertical and the other side is at negative 45 degrees. You can detect the difference between one side and the other by placing a piece of polarizing film in front of a phototransistor or photoresistor.

5.5.7 Reflectance Sensors

A reflectance sensor is made up of a combination of an infrared or red LED and a phototransistor that is sensitive to the wavelength of light being emitted by the LED. Over dark surfaces, the light is absorbed, whereas over light surfaces, the light is reflected back to the phototransistor.

A reflectance sensor (figure 5.9) can be made using discrete components.

The reflectance sensors are useful for detecting what color the floor is. They could also be used as object detectors, but they are very near sighted and quite responsive to outside lighting. In any application, good shielding is an absolute requirement if any reliability is desired. The sensors are very sensitive to distance from the reflecting surface. Distances greater than an inch will give very poor reading, and distances that are too small will not allow the right to be reflected. The angle at which the light is reflected to the surface is important and can produce better or worse results at different distances.

5.5.8 Motor-Force Sensors

There are four motor force sensors built into the 6.270 Controller board, attached to motors 0 through 3. These sensors are included to detect when the motors might be stalled. When the motors stall, they draw a large amount of current, which appears as a large voltage on the analog inputs to the 6811. When a motor_force value increases sharply, that's a good sign, but not guaranteed, that the motor may be stalled. The value that it reaches will depend on the load attached to the motor. Experiment by stalling the motor yourself while printing the values on the LCD to determine a threshold that's right for your robot.

Motor force values are not very accurate when you are driving the motors at anything less than 100%. Driving the motors at lower speeds will cause the motor force value to oscillate wildly, so it is recommended that you only use this information when you are driving a motor at full speed.

5.5.9 Breakbeam Sensors

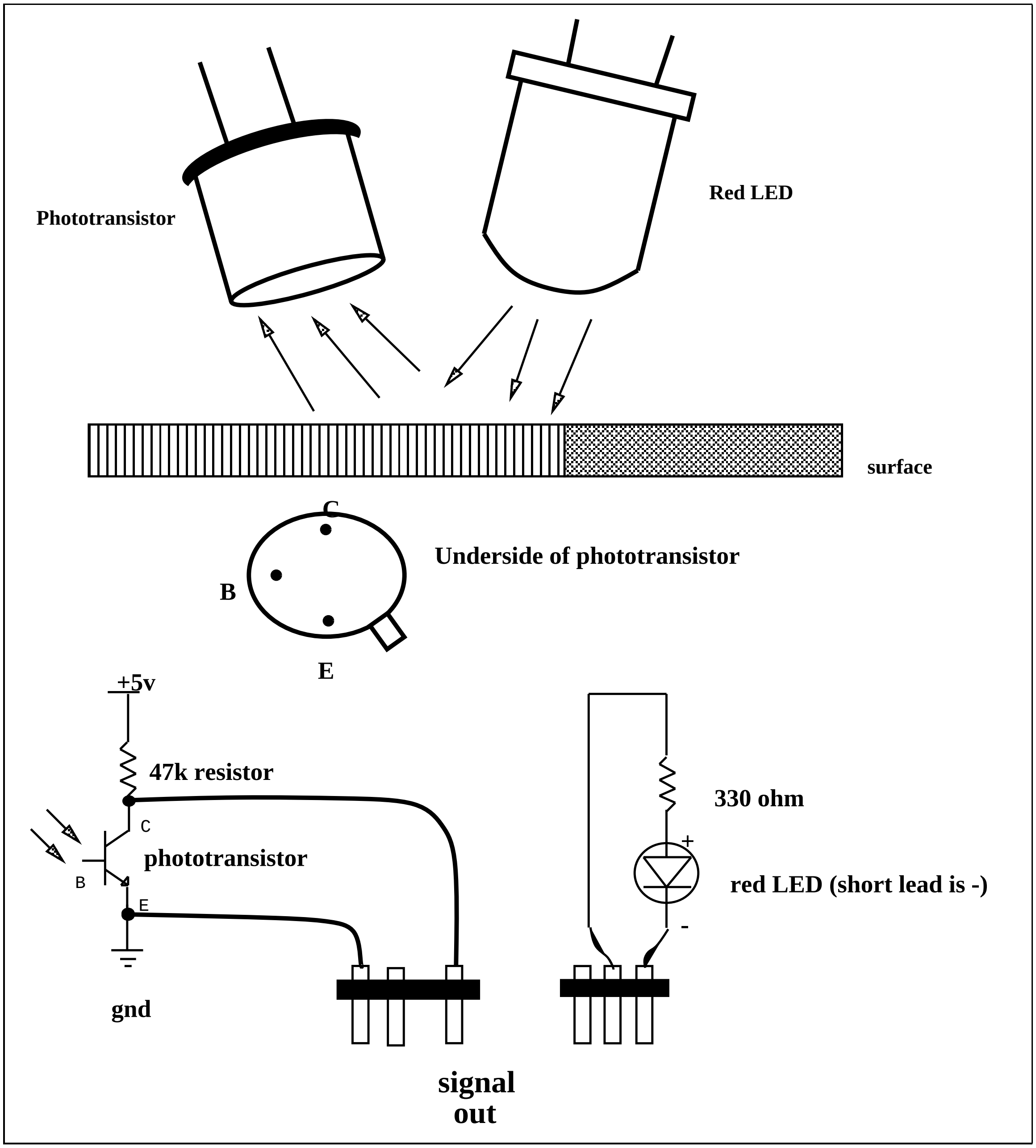

Figure 5.9: Reflectance Sensor

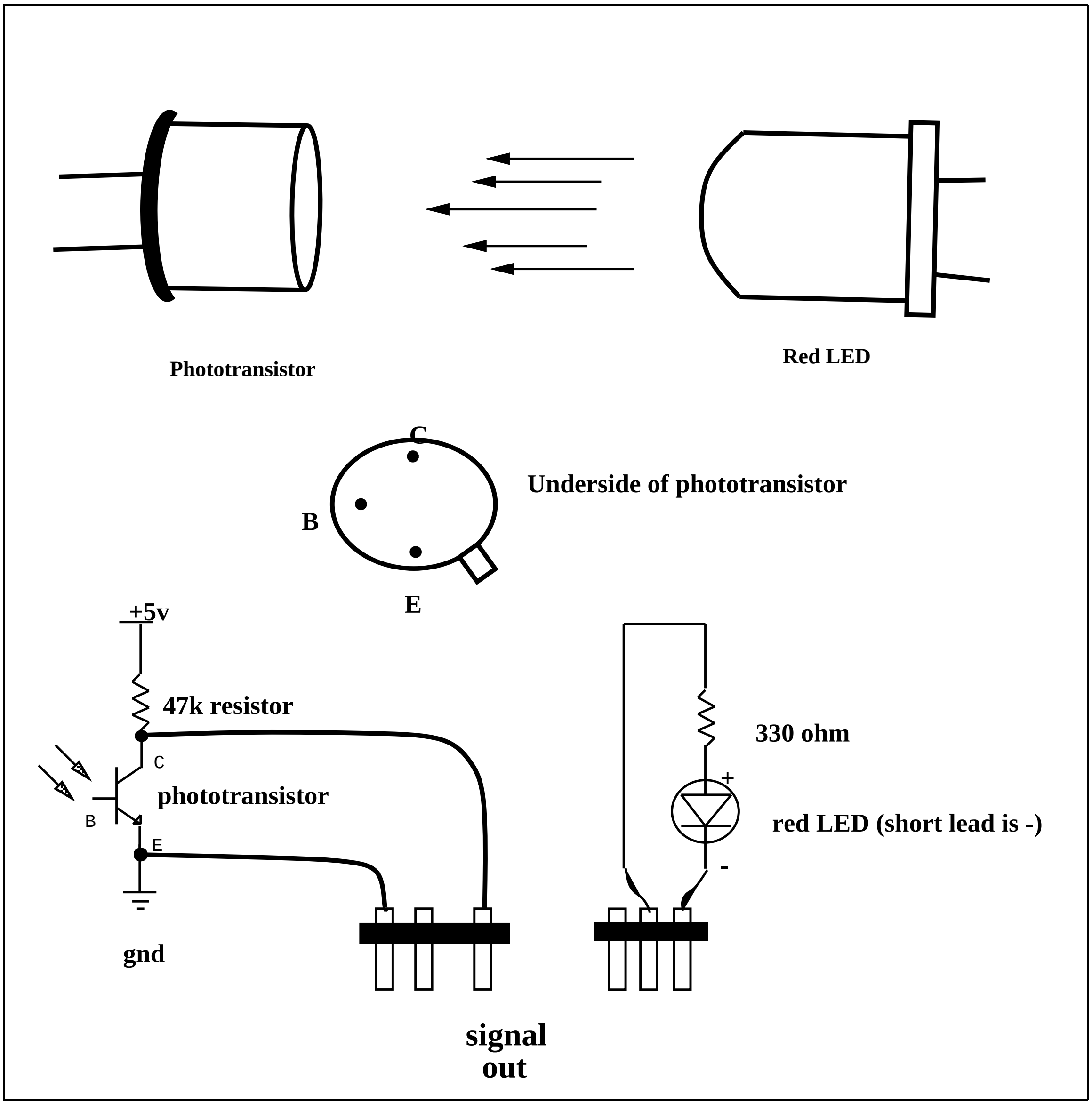

Figure 5.10: Breakbeam Sensor using discrete components.

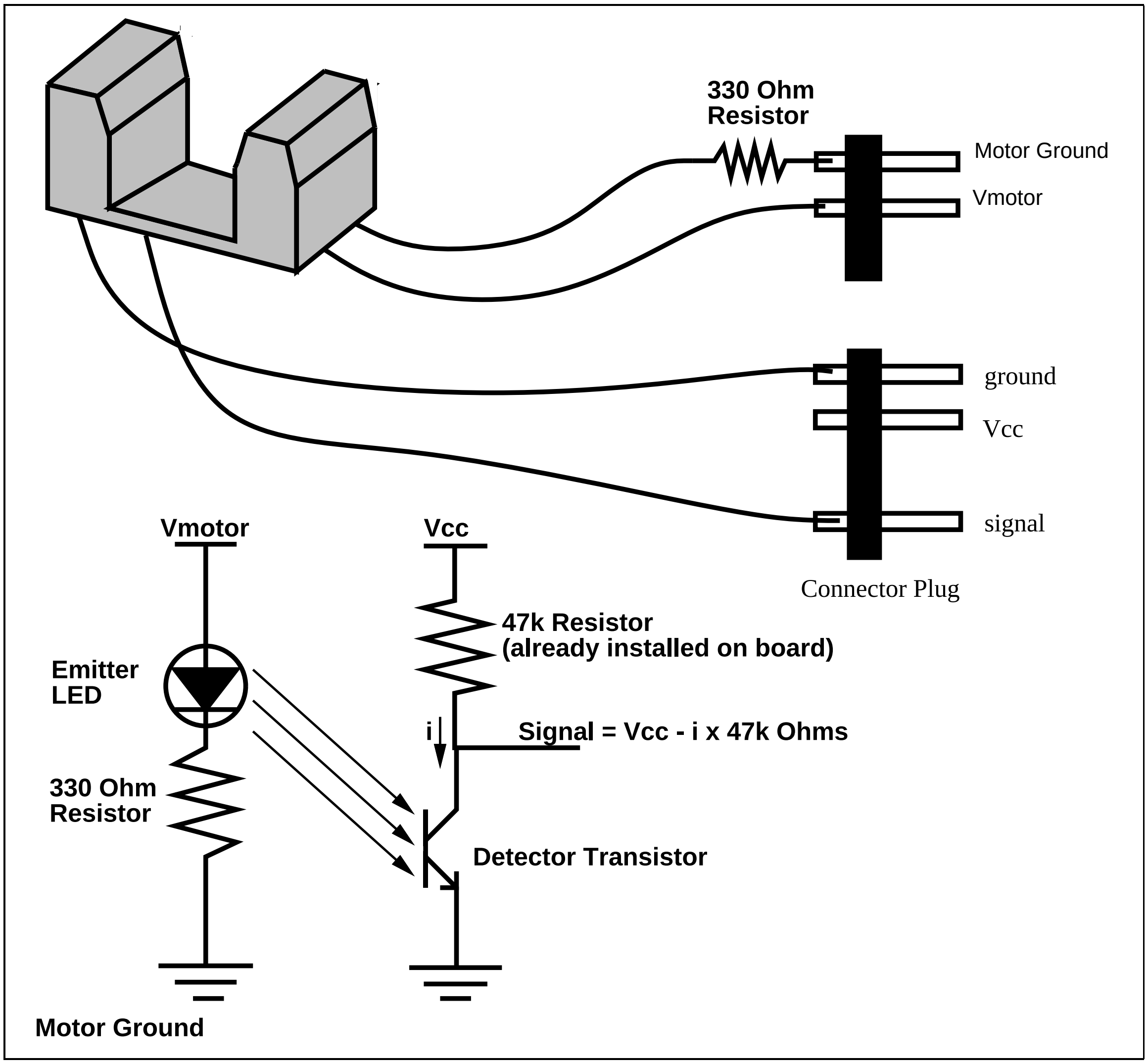

Figure 5.11: Breakbeam Assembly

Breakbeam sensors are another form of light sensors. Instead of looking for reflected light, the photosensor looks for direct light as shown in Figure 5.10.

The sensor is useful in detecting opaque objects that prevent the light beam from passing through. This can be useful in detecting block between gripper, or when block passes through a passageway. The sensor does not need to detect the block very quickly so the phototransistor can be plugged into the analog port.

The breakbeam sensors can also be used for counting holes or slots in a disk as it rotates (see Figure 5.12), allowing distance traveled to be measured. Since this requires a very fast sampling, the sampling needs to be done at the assembly language level. We have implemented shaft-encoder routines to do the fast sampling. But in order to use these routines the sensors should be plugged into the lower two digital ports if the rate at which the holes or slots go by is very high.

Before you use the analog sensors in the digital switch you must make sure that there is a full swing in the analog reading from when the light goes through to when the light is blocked.

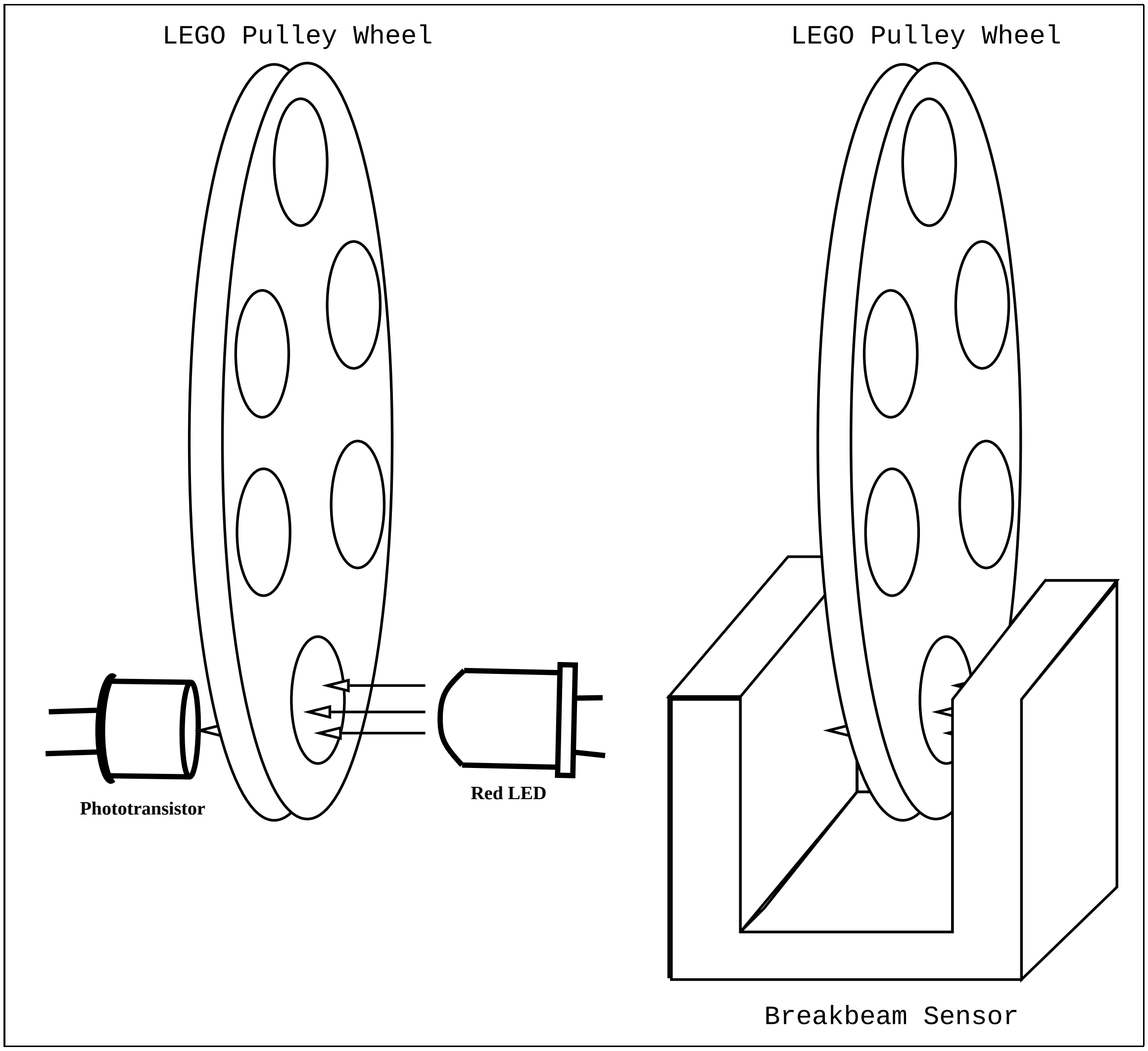

Figure 5.12: Shaft encoding using a LEGO pulley Wheel