Photoresistor, Transistor, and LED's

Learning Objectives:

By the end of this laboratory experiment, the experimenter should be able to:
- Explain how a photoresistor works
- Describe the voltage-current relationship for an LED
- Build a circuit that includes an LED, photoresistor, and transistor and interface the circuit to a microcontroller to create a light controlled switch
- Write a program for the ATmega16 to control and modify the functionality of a light controlled switch

Components:

| Qty.	Item	Qty.	Item
2	10 kΩ resistor	1	470 Ω resistor
1	Solderless Breadboard	1	220 to 270 Ω resistor
1	Photoresistor	1	red or green LED
1	2N3904, NPN transistor		
1	STK500 with Atmel ATmega16		

Introduction:

A photoresistor is simply a resistor whose resistance depends on the amount of light incident upon it. Photoresistors are used to make light-sensitive devices, and are often made from cadmium sulfide (CdS). The resistance of a CdS photoresistor varies inversely to the amount of light incident upon it. In other words, its resistance will be higher at low light levels (in the dark) and lower at high light levels (in the light).

A light emitting diode (LED) behaves like an ordinary diode except that when it is forward biased, it emits light. An LED's forward voltage drop is higher than that of an ordinary diode. Typical LEDs (the two-wire leaded "jelly bean" type) require 5 to 15mA to reach full brightness, but are not designed to handle more than about 20 mA of current (though some designed specifically for lighting applications can handle upwards of 1A or more). You will therefore *always* need to provide a resistor in series with an LED to limit the current to about 20 mA or less, or else you will burn it out. Also, don't make the mistake of trying to substitute an LED where a standard diode is called for! Look at the schematic diagram to see which kind of component is needed.

Procedure

1. Measure the photoresistor's resistance in the ambient lighting of the lab. How stable is it? Once this is recorded, repeat the measurement, only this time, cover the cell with your hand. These two extremes will be used in calculations later on.

2. To verify the behavior of the LED, construct the circuit shown in Figure 1. Measure the actual resistance of the 470 Ω resistor and record your reading. Vary the supply voltage from 1 to 5 volts using 1-volt increments. **At each voltage, measure the voltage across the LED and the 470 Ω resistor using the DMM, and enter the values into the following table.**

The LED current can be calculated by applying Ohm's law across the resistor. A similar table should be entered into the lab report with all voltage values and comments. Remember Ohm's law for calculating the current through a resistor: $I = V/R$, where V is the voltage *across* the resistor.
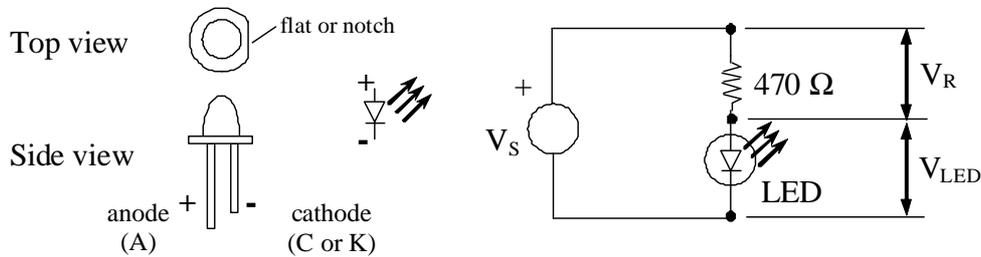
Figure 1. LED and typical circuit. Note that the anode lead is longer than the cathode. Sometimes there is a flat on the cathode side of the LED to help you distinguish anode from cathode. With voltage sources above the maximum forward voltage of the LED, *you **must always** use a resistor in series to limit the current through the LED*.

Table 1. LED circuit measurements (Refer to Figure 1)

V_S, V	V_{LED}, V	V_R, V	Current, mA	Comment on LED brightness
1				
2				
3				
4				
5				

Figure 2 shows a simple 'light-controlled-LED'. The circuit should turn-off the LED as the photoresistor is covered. **Explain the theory of operation of this circuit**. Based on the information obtained above, what is a good supply voltage to use? (Hint: V should be high enough so that enough current flows through the LED when the photoresistor has low resistance, and yet should be low enough so that the current is not enough to turn on the LED when the photoresistor has high resistance.) **Build the circuit in Figure 2, and check its function. Describe its operation.**
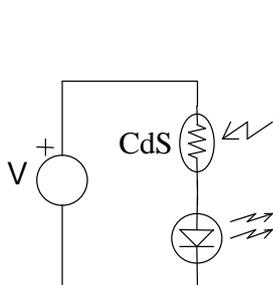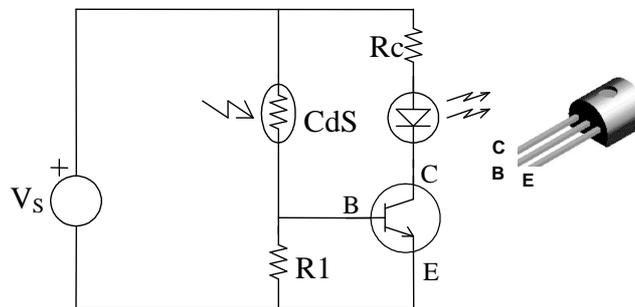
Figure 2. Light-controlled LED
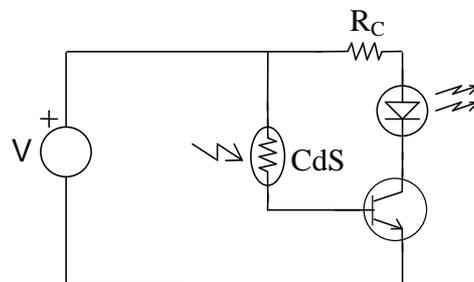
Figure 3. Light-controlled using a transistor "switch"

The Light-Controlled Switch Using a Transistor

A transistor can be added to the light-controlled-switch circuit to improve its sensitivity and to eliminate the 'half-on-half-off' state of the LED. A rudimentary circuit to do so is shown above in Figure 3 **(you don't have to build this circuit).** Here the photoresistor controls the amount of current flowing into the base of the transistor, which in turn controls the collector current of the transistor, thus controlling the current through the LED. Unfortunately, this circuit may not function properly, because when the photoresistor is in the dark state, and the LED is supposed to be turned off, the base current may be large enough that the LED may stay lit! **Prove this (not now, but when you write your report), by calculating the collector current for the circuit in Figure 3 when V=10 V, R_{CdS}=100 kΩ, R_c=220 Ω and h_{fe}=100.**

Figure 4 shows an improved circuit. This is the circuit that you will build and experiment with next.

Figure 4. Improved light-controlled switch using a 2N3904 transistor. With the flat side of the transistor facing away from you, the pins from left to right are: collector, base, and emitter. (Transistor pinout source: http://www.fairchildsemi.com/ds/2N/2N3904.pdf)

With a properly selected resistor R_1, the voltage at the base of the transistor in the dark state is less than 0.7 V, and therefore the transistor is in the cut-off state. Since the transistor is cutoff, no current flows from its collector to its emitter, so the LED will be off. As the photoresistor's resistance decreases (as the result of an increase in light intensity), the voltage at the base increases due the voltage divider formed by R1 and the photoresistor. Once the base voltage reaches 0.7 V, the base current starts to flow, and any further decrease in the photoresistor's resistance causes an increase of base current. This base current increment will be amplified by the current gain of the transistor up to the point that the transistor saturates.

Procedure

The following procedure explains how to select the resistance values for R1 and R_C in Figure 4. **Document with calculations in your report how you arrived at the resistance values that you ultimately used to build the light-controlled transistor switch in Figure 4 using the procedure below.**

1. **Choose the supply voltage, V_S.** The supply voltage is often a predetermined value rather than a design choice. For example, suppose the supply consisted of five alkaline batteries in series. Then V_S = 5 * 1.5V = 7.5V.

2. **Select R1**. First, choose the value of the photoresistor's resistance (call it R_{on}) within the range that you measured earlier at which you would like the LED to be turned on. The resistance value can be that for when the photoresistor is covered or somewhat "shaded", it's up to you. The value of R1 should be such that **0.7V = V_S*R1/(R1+R_{on}).**

Choose a nominal resistor value closest to the value you calculated for R1, or you can use a variable resistor (trim pot). Depending on the resistor value you choose, you may want to check the equation above for R_{ON} to make sure it still within the range that you want.

3. **Select R_c,** the current limiting resistor. With this resistor, the collector current is limited to $I_{max}=(V_S - V_{LED} - V_{Sat})/R_C$, where V_{LED} is the voltage drop across the LED (i.e., the voltage you measured across the LED when it was 'on'. See Table 1 above.), and V_{Sat} is the saturation voltage between the collector and emitter. A reasonable value for V_{Sat} for the 2N3904 transistor is about 0.4V. **Select R_C so that the LED current is limited to be less than 20 mA (preferably 5-10 mA).** You may need to iterate this calculation depending on the limited choice of resistor values that we have in the lab. Using R_c, **construct and test the circuit**.

Using the ATmega16 to Make a Programmable Light-Controlled Switch

The circuit in Figure 4 is very simple, but it suffers from the disadvantage that once R1 is chosen and the circuit is constructed, you're stuck with its performance unless you physically remove R1 and replace it with a different value. That is not too serious if you are dealing with one circuit on a breadboard, but suppose this circuit were part of a product that you were manufacturing, say 1000 per day. If you wanted to change the performance of the device, you would have to modify the assembly drawings, circuit board artwork, component inventory, rework the entire work-in-process, etc. That would be a big deal! Here we will use the ATmega16 to make a light-controlled switch whose performance can be modified by simply _reprogramming_ the microcontroller.

Procedure

1. Build the part of the circuit in shown in Figure 5 below labeled A. We are going to use the STK500 to supply power and ground to this circuit for convenience, but don't assume that you can power do so as a general approach to your prototyping needs! Loads that draw more than a couple tens of milliamps should be powered by an external supply with a common ground to the STK500. Connect the 10k resistor to pin PC0 on the STK500 board using a jumper from the pin to the solderless breadboard. Don't forget that you need to supply power to the ATmega16. The voltage on PC0 will either be 0 or 5 V. **Which of these will turn the transistor on? When the transistor turns on, is it saturated? Based on your earlier measurements on the LED and the characteristics of the transistor, what are i_b, and i_c when PC0 is at 5 V?**

2. Create a New Project in AVR Studio. Refer to the previous lab if you don't remember how to do this. Also, don't forget to set the Configuration Options with the CPU frequency at 8 MHz, O2 optimization, and choose the proper directory location of the hex file. Compile and run the **Blink Test Program**, which you can find on the ME 106 website, and see if you can make the LED blink.

   Explain how the PORTC output is toggled by using the '^=' operator.

3. When you have _successfully_ completed step 2, build the circuit in B shown in Figure 5. Create a New Project, make copies of uart.h and uart.c from the ME 106 website, and compile and run the **Photoresistor and A/D test program**, which you can also find on the ME 106 website.

   Connect the microcontroller to Hyperterminal as described in the Introduction to ATmega16 lab. Reset the microcontroller by pressing the reset button on the STK500 or by cycling the power. **What happens when you cover the photoresistor? What range of values are shown in the communications window? What voltage do these correspond to?**
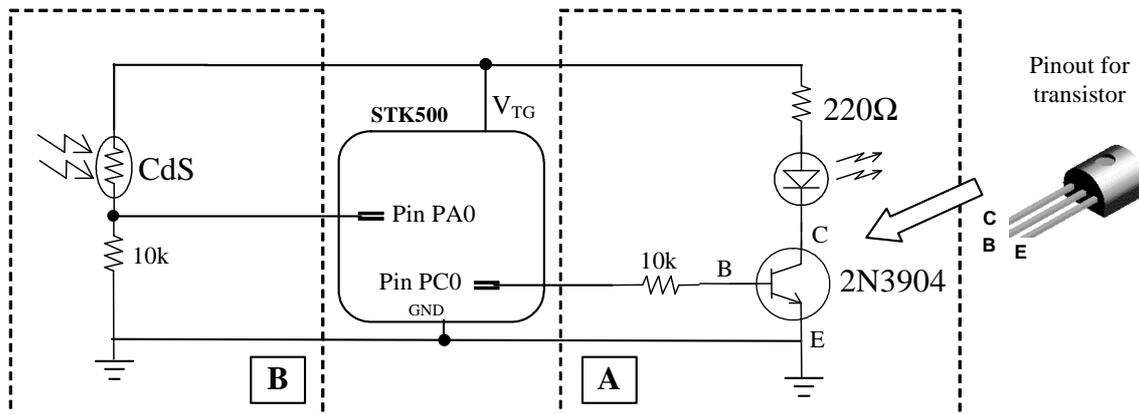
Figure 5. Light-controlled switch using the ATmega16. The photoresistor (in B) is part of a voltage divider, the output of which is connected to pin PA0, one of the pins that can be used for analog-to-digital conversion. Pin PC0 is connected to the base of a transistor (in A), which is used as a switch for illuminating the LED when the light level on the photoresistor is beyond a certain threshold value. (Transistor pinout source: http://www.fairchildsemi.com/ds/2N/2N3904.pdf)

4. When you have *successfully* completed step 3, write a program that will turn the LED on when you cover the photoresistor with your hand. (Hint: add a test in the While loop that compares the value of adc_level with a value approximately in the center of the voltage range that you measured in step 3 when the photoresistor was covered. Experiment with the threshold value.) Display the state of your LED on the terminal window.

What changes need to be made to the software (note: no need to change any *hardware*) if you want to have the LED stay on under ambient light conditions and turn off when a shadow falls on the photoresistor (i.e. the opposite function to what you programmed in step 4)?