

STEMSEL Beginners Project 8: Smart Light Dimmer

Problem

We want to control the voltage going to a lightbulb in order to make a dimmer light. We also want this light to sense the brightness of the room and dim itself automatically to save power if the room is bright.

Background

While many lights just turn on or off, some lights allow you to control the brightness. This is useful since it allows you to adjust how much light you want, and can also save electricity since if you only need the light at half brightness it will use less electricity than if it was full brightness.

This project uses the controller board to control the amount of light falling on a light sensor. A feedback loop maintains the amount of light detected by the light sensor even if the lightbulb is moved.

The amount of light emitted by the globe is controlled by the potentiometer by changing the on/off duty cycle at a high frequency. Duty Cycle means the percentage of ON-Time compared to the Period (On-Time and Off-Time). The period of the signal stays the same, so if the On-time is longer, the Off-time must be shorter, and vice versa. This effectively changes the voltage that the lightbulb sees; if the duty cycle is high then the voltage seems high, and vice versa. This is called Pulse Width Modulation (PWM).

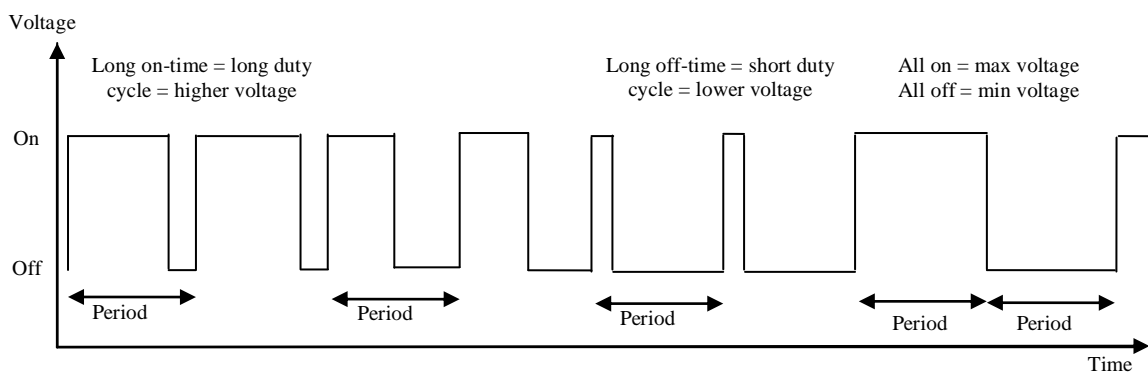


Figure 1

This is related to the previous musical buzzer project, since once again we are controlling the wave to affect the output, but instead of changing the period, we are keeping the period the same and only changing the percentage of on-time. Remember that for the buzzer, the on-time and off-time were always equal.

Ideas

How can we create our PWM signal? How can we control the length of the duty cycle? How do we ensure the period is always the same length and how long will it be? What components will we need to solve the problem?

Plan

To solve the problem we can use the STEMSEL controller board, a light globe, a Light Dependent Resistor and a potentiometer. The POT will be used to control the length of the duty cycle (on-time). We know that the POT can be used to set a number between 0 and 255, so 255 will be period. Therefore, the duty cycle will be set by the POT, and the off-time should be the period minus the on-time, i.e. $255 - \text{POT value}$. So if the POT is set to 0, the on-time is 0 and the off-time is $255-0=255$. If the POT is set at 255, the on-time is 255 and off-time is 0. If it is between these values, for example 210, then the on-time will be 210 and the off-time will be $255-210=45$.

The LDR can also be used to adjust the duty cycle, such that if the room is already bright the microchip can shorten the duty cycle to dim the light. So the microchip needs to get the length of the duty cycle by checking the POT and/or LDR, then send the correct PWM signal to the lightbulb.

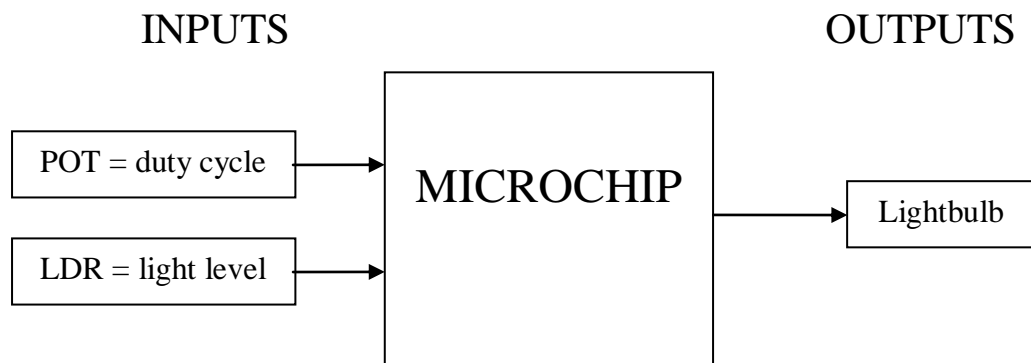


Figure 2

Design the Circuit

Use ezCircuit Designer to construct an input/output (I/O) diagram. We will need one driver, one LDR and one potentiometer. Right click to rename the icons.

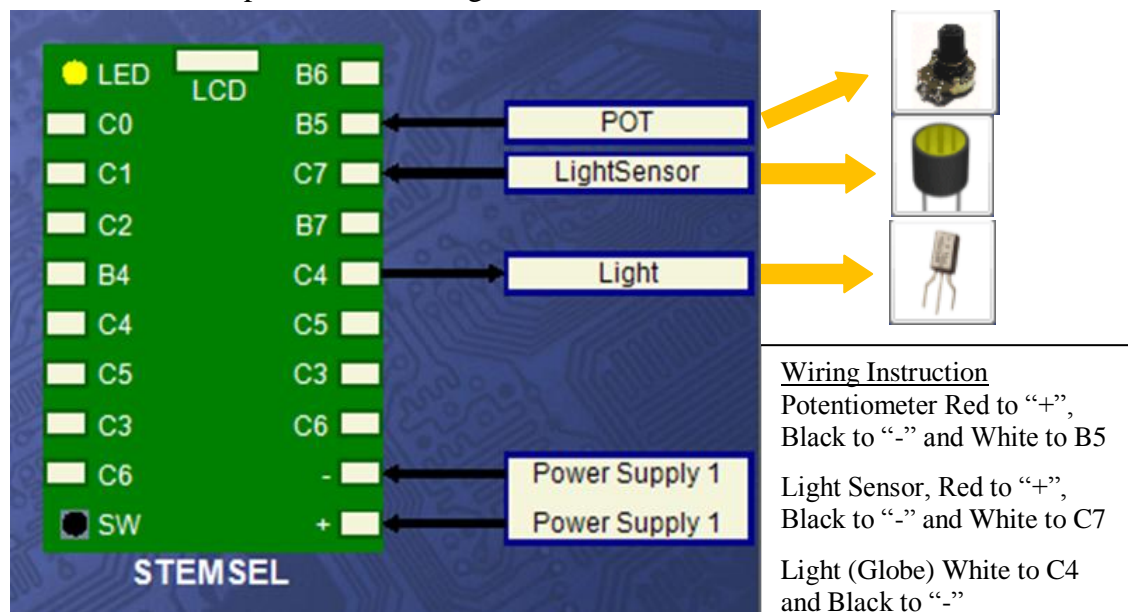


Figure 3

Build the Circuit

Use the Circuit Designer I/O diagram to connect the hardware.

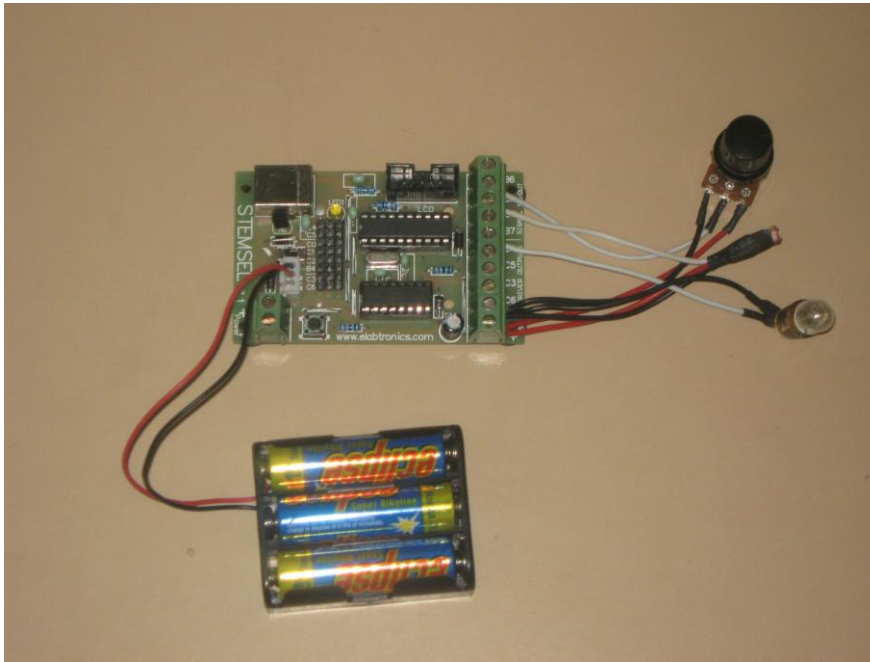


Figure 4

Programming

Use CoreChart to program the functions of the smart light dimmer. After you have used the test routines to check the outputs, delete them so that we can start writing our own program.

1. According to the plan, the first thing we need to do is set our duty cycle. Use an AnalogIn icon to save the value from the POT as OnTime
2. Recall that in the plan we said the period of our PWM signal would be 255, since our POT can vary from 0-255. Use a Subtract icon to subtract OnTime from 255 and save the result as OffTime.
3. Use an OnOffPin to turn the light on.
4. Add a TimeDelay icon, but instead of typing in a fixed value we are going to use the OnTime value set by the POT. Ensure you click the little arrow in the ten thousandths box to select OnTime.
5. Using another OnOffPin and time delay, turn the light off for 'OffTime' ten thousandths of a second. Because OnTime + OffTime is always 255, this ensures our period stays the same even if OnTime and OffTime change.
6. Finally, add a GoTo START icon so the program loops continually. The finished program should look like this. Send the program to the chip, then try turning the POT to adjust the light brightness.

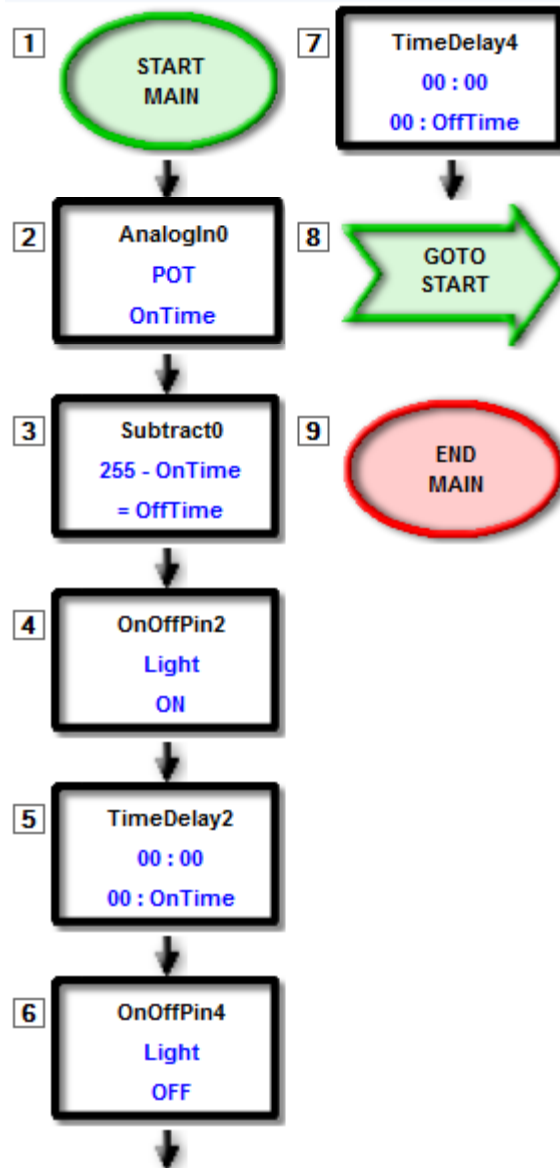


Figure 5

Extension

Use the Light Dependent Resistor to reduce the brightness of the smart light to save power. Modify the smart light dimmer program so that the light sensor is also used to control the brightness of the light. The globe brightness will be reduced when there is a high level of ambient light and the brightness will be increased when there is a low level of ambient light.

Hint: $\text{OnTime} = \text{OnTime} - \text{Ambient Light Level from LDR}$

Summary

PWM is a useful technique used for turning things only partway on. It can be used for many applications, such as to control the speed of a motor or the brightness of a light like in this project. In this way it both gives us more control and saves electricity.