Arduino Sustainability Project

Prepared for:
Dr. Pushkin Kachroo
EE 290
UNLV

Christian Calvo
2009-04-31
Introduction
This project was intended to come up with a device, which is controlled by the Arduino microcontroller to satisfy the theme of sustainability. The device I decided to construct is a simple digital alarm clock display that would shut off the display to save energy.

To simplify the design, I chose to only use one 7-segment display to simulate the clock. In addition, to simulate the time, the display of the 7-segment display was advanced every second by the program that was uploaded to the microcontroller.

The original design was going to be controlled by a 7-segment decoder/driver, in which the display is shut off between two hours that would be set by the user, but I was unable to find this IC in a store. Due to this hindrance, I revised my design to contain a light dependent resistor (LDR). This change in design proved to be an improvement because now the device requires no user input and is fully automated. The clock display is still controlled by the microcontroller, however the power to the display is determined by the LDR. The LDR output signal is fed into an analog input of the Arduino. The value is read by the microcontroller, in which it is compared to a value that can be changed to adjust the sensitivity. The device setup and schematic are shown below.
As you can see from the schematic above, Digital Pins 2 through 8 are the signal sends to the 7-segment LED display and Digital Pin 12 is the power send to the display. The Analog Input 0 is used to read the voltage drop across the light dependent resistor. That value is then compared to a value that is set in the program. If the value is greater than the reference value, Digital Pin 12 is set to HIGH. If the value is below the reference, Digital Pin 12 is set to LOW.
Appendix

```cpp
int light;
void setup()
{
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(12,OUTPUT);
  light = analogRead(0);
  Serial.begin(9600);
}

void loop()
{
  light = analogRead(0);
  Serial.println(light);
  if (light > 35)
  {
    digitalWrite(12,HIGH);
  }
  else
  {
    digitalWrite(12,LOW);
  }
  digitalWrite(2,LOW); //0
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
delay(1000);
light = analogRead(0);
Serial.println(light);
if (light > 35)
  {   digitalWrite(12,HIGH);
  }
else
  {   digitalWrite(12,LOW);
  }
digitalWrite(2,LOW); //1
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,HIGH);
delay(1000);
```
light = analogRead(0);
Serial.println(light);
if (light > 35)
{
digitalWrite(12,HIGH);
}
else
{
digitalWrite(12,LOW);
}
digitalWrite(2,LOW); //2
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
delay(1000);
light = analogRead(0);
Serial.println(light);
if (light > 35)
{
digitalWrite(12,HIGH);
}
else
{
digitalWrite(12,LOW);
}
digitalWrite(2,LOW); //3
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
delay(1000);
light = analogRead(0);
Serial.println(light);
if (light > 35)
{
digitalWrite(12,HIGH);
}
else
{
digitalWrite(12,LOW);
}
digitalWrite(2,LOW); //4
digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,HIGH);
delay(1000);
light = analogRead(0);
Serial.println(light);
if (light > 35)
{
    digitalWrite(12,HIGH);
}
else
{
    digitalWrite(12,LOW);
}
digitalWrite(2,HIGH); //5
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
delay(1000);
light = analogRead(0);
Serial.println(light);
if (light > 35)
{
    digitalWrite(12,HIGH);
}
else
{
    digitalWrite(12,LOW);
}
digitalWrite(2,HIGH); //6
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
delay(1000);
light = analogRead(0);
Serial.println(light);
if (light > 35)
{
    digitalWrite(12,HIGH);
}
else
{
    digitalWrite(12,LOW);
}
digitalWrite(2,LOW); //7
digitalWrite(3,LOW);
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,HIGH);
delay(1000);
light = analogRead(0);
Serial.println(light);
if (light > 35)
{

digitalWrite(12,HIGH);
}
else
{
digitalWrite(12,LOW);
}
digitalWrite(2,LOW); //8
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
delay(1000);
light = analogRead(0);
Serial.println(light);
if (light > 35)
{
digitalWrite(12,HIGH);
}
else
{
digitalWrite(12,LOW);
}
digitalWrite(2,LOW); //9
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,HIGH);
delay(1000);
light = analogRead(0);
Serial.println(light);
if (light > 35)
{
digitalWrite(12,HIGH);
}
else
{
digitalWrite(12,LOW);
}
digitalWrite(2,LOW); //reset
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
}