Dineth Hettiarachchi
CSE 3323

Smart Lamp

Product Concept

The main purpose of the Smart Lamp is to wakes up a person gradually by simulating the sunrise. Therefore, it is possible to pre-set the alarm and when the alarm goes off, the LEDs gradually increase their brightness (white color) and will stay at full brightness for a while until they turn off and resets the alarm. In addition to this feature, the lamp also provides the night lamp functionality by automatically turning on the LEDs at a lower brightness level (blue) when the surrounding environment is dark. Also, if the device detects any movement during that period, the LEDs change their color to green at full brightness.

Product Outline

The circuit of the Smart Lamp can be divided into 5 parts.

1. Power supply
2. Standalone circuit for ATmega 328
3. Ambient light detector
4. Motion detector
5. RGB LED driver

Bill of materials

1. 1x 12V power adapter
2. 1x slide switch
3. 1x resettable fuse
4. 1x 100µF capacitor
5. 1x 10µF capacitor
6. 1x rectifier diode (1N4001)
7. 1x voltage regulator (LM7805CT)
8. 1x Arduino Uno board (for development purpose only)
9. 1x USB Cable (for development purpose only)
10. 1x ATmega 328 chip

11. 2x 10K resistors
12. 1x 100Ω resistor
13. 1x 16 MHz crystal
14. 2x 22pF capacitors
15. 1x Mini push button switch
16. 1x DIP sockets solder tail 28-pin 0.3”
17. 1x Photocell
18. PIR Movement Sensor (HC-SR501)
19. 3x MOSFET N Transistors (STP16NF06FP)
20. 1x 12V LED strip
21. 1x A Cylindrical Shape Can
22. 1x digital multimeter (for development purpose only)
23. Hook-up wire
24. 1x soldering iron
25. Solder
26. Wire strippers, wire cutters
27. 1x prototyping board

Power supply

The LED strip needs a 12V DC source while the Arduino/ATmega 328 requires a stable 5V DC source. Therefore, a 12V DV power adapter that can provide 1A was chosen for this project. To covert the 12V to 5V a voltage regulator (LM7805) was used. The rectifier diode protects the circuit from connecting the power adapter incorrectly. A slide switch is used so that the device can be turned on/off easily. PTC/resettable fuse breaks the connection if the circuit draws more than 1A. Since the power adapter converts the AC current to DC current using a transformer, the output current comes with signal noise. Therefore, the 100µF and 10µF capacitors are used to minimize the signal noise by smoothing out dips and rises of the signal.

Dineth Hettiarachchi
CSE 3323

Standalone circuit for ATmega 328

First the program needs to be uploaded to the ATmega chip using the Arduino Uno board and a USB cable. Then the DIP sockets solder tail 28-pin 0.3" was soldered into the prototyping board so that the ATmega chip can be removed from the circuit and install into the Arduino board incase if the program needs to be re-uploaded. The ATmega 328 has two pins (pins 8, 22) that need to be connected to the ground. The +5V source from the power supply earlier needs to be connected to the pins 7, 21, 22. Connecting a pushbutton switch and a 10KΩ resistor to pin 1 as indicated in the schematic diagram would allow us to reset the state at any given time. As most of the microcontrollers, ATmega 328 also needs a clock in order to work. Therefore, the 16 MHz crystal was chosen as it would be compatible with the chip itself. Also, the 22pF capacitors helps the crystal to start oscillating.

Ambient Light *Detector*

A photo resistor/photocell is used along with a 10KΩ resistor (used as a voltage divider) to detect the ambient light of the room. The output of the photocell is connected to the pin 23/analog input 0.

Motion Detector

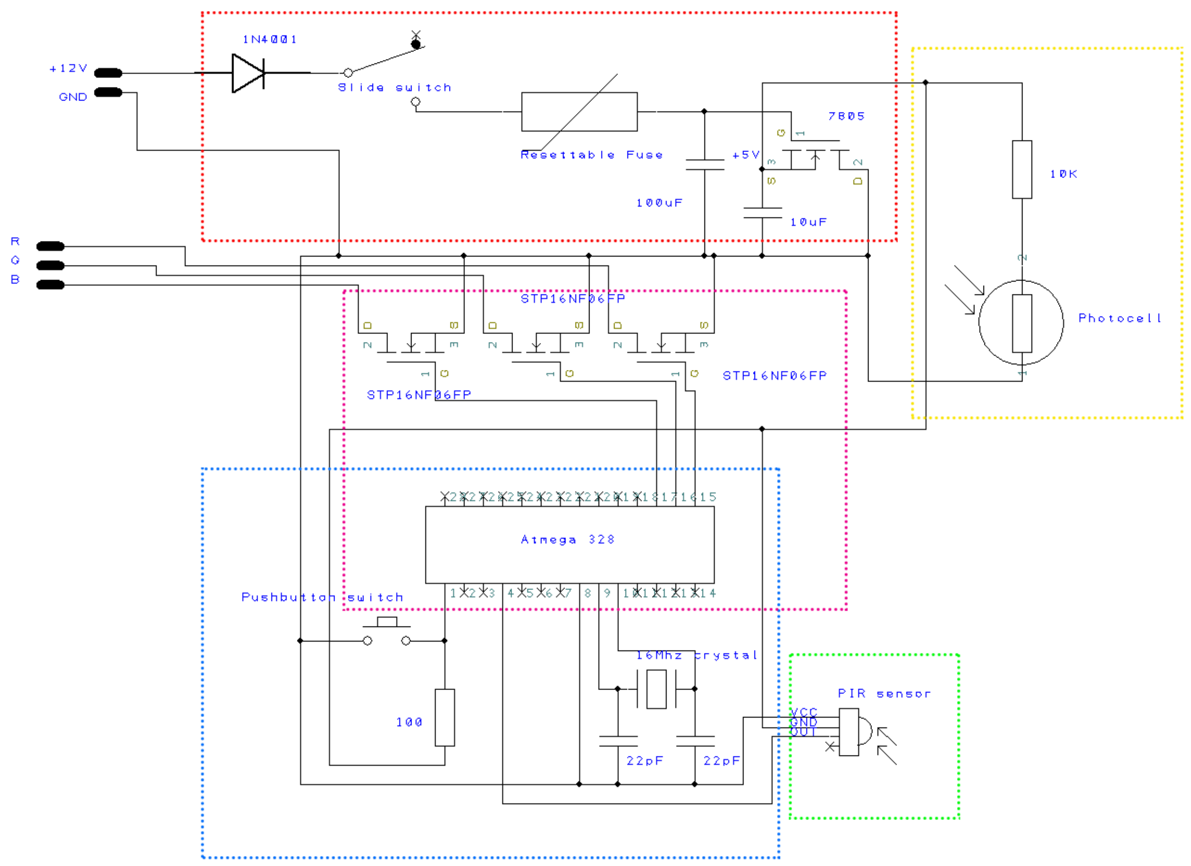
The PIR motion sensor requires a +5V source and needs to be connected to the ground. The output of the PIR unit is connected to the pin 4/digital pin 2 or the ATmega 328.

RGB LED Driver

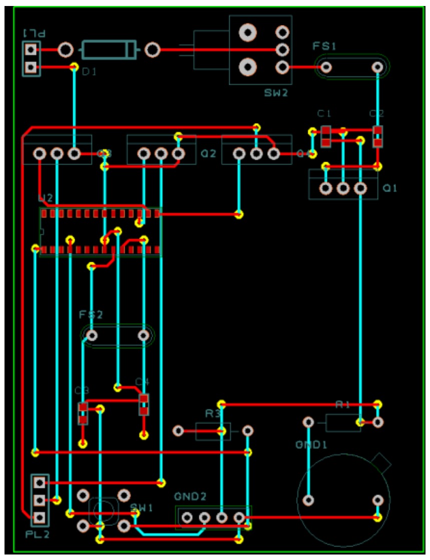
The LED strip requires a +12V source. Therefore, it is connected to the +12V source directly. 3 MOSTFET N (STP16NF06FP) were connected to the LED strip and the pin 15, 16, 17 (G, B, R) so that the voltages for the RGB channels can be altered using PWM to get different color combinations.

Dineth Hettiarachchi
CSE 3323

Schematic Design

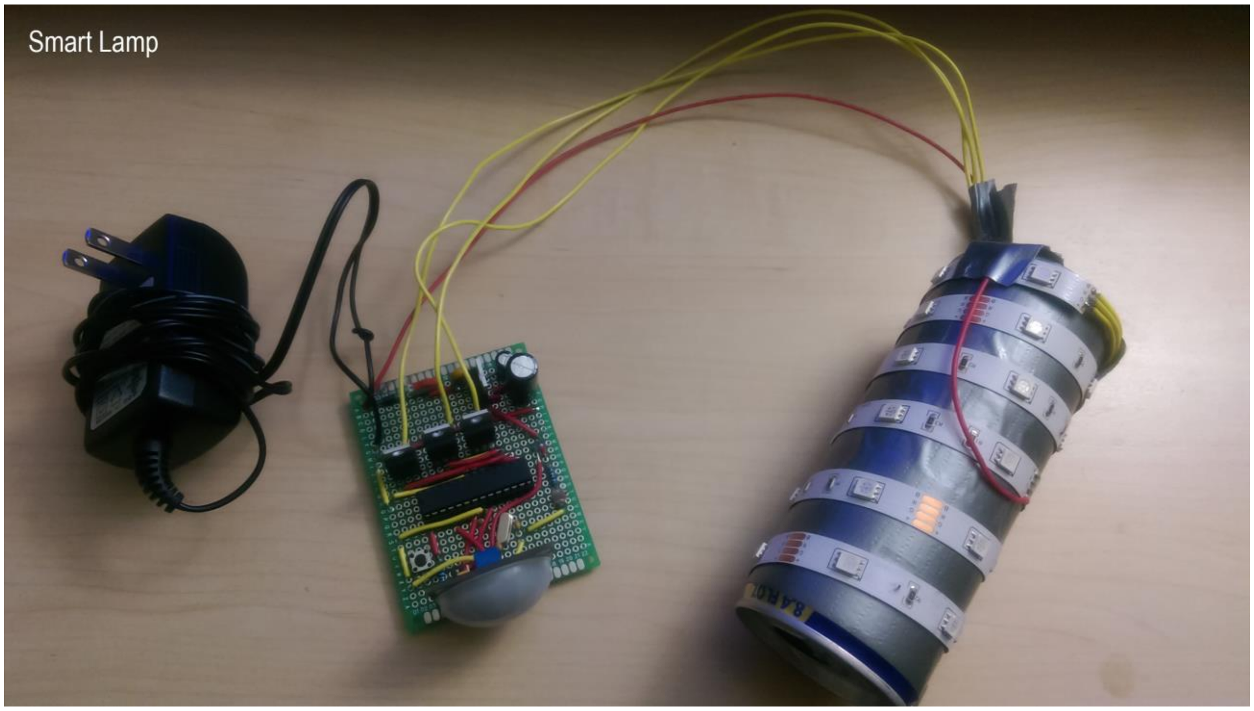


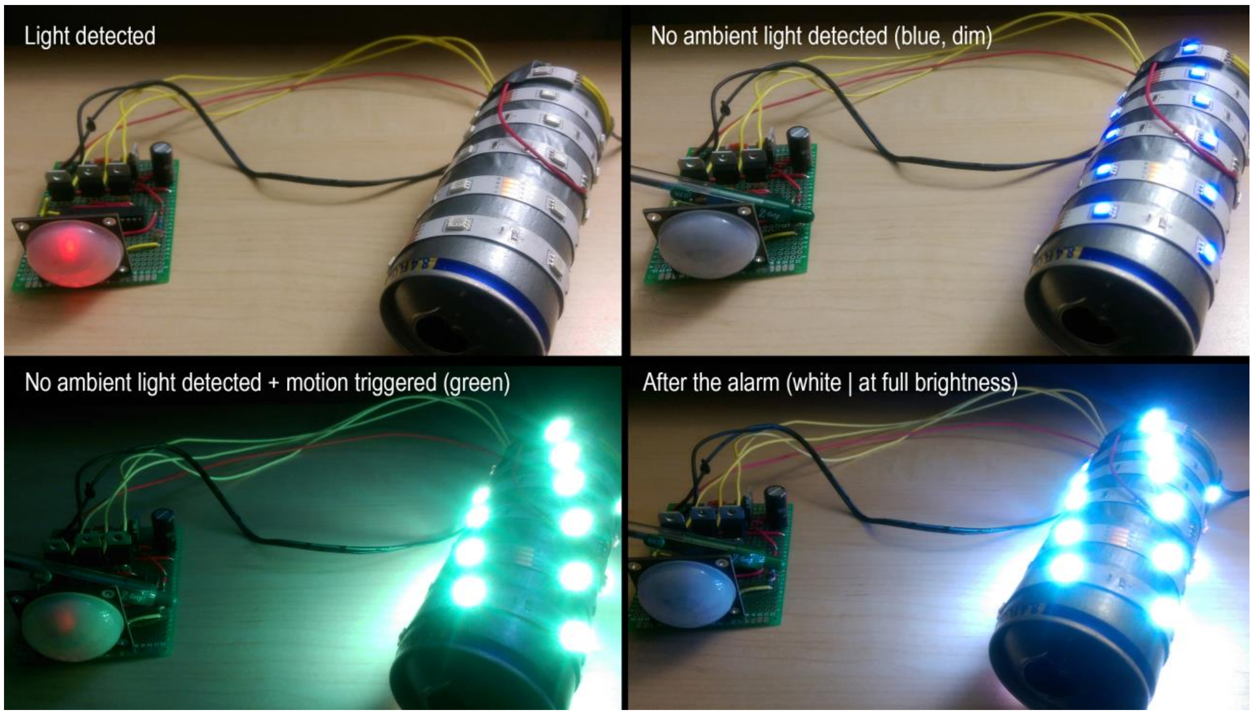
PCB Design



Dineth Hettiarachchi
CSE 3323

Prototype




Light detected
No ambient light detected (blue, dim)
No ambient light detected + motion triggered (green)
After the alarm (white | at full brightness)

Demo: https://www.youtube.com/watch?v=bcLRLUIvyWE

Dineth Hettiarachchi
CSE 3323

Arduino Sketch

```
/**
  Smart Lamp
  **/
/* How many milliseconds the nightlight should stay on for (1 minutes would be 1 x 60 x 1000ms = 60000 */
int nightLightTimeOut = 60000;
#define RED 9
#define GREEN 11
#define BLUE 10
#define nightLight 13
#define trigger 2
unsigned long currentMillis = 0; // stores current time value
unsigned long previousMillis = 0; // the value of the previous days millis(), so we can reset each day
unsigned long currentMinutes = 0; // just easier to work with
unsigned int minutesUntilSunrise;
unsigned long nightLightTimeOff = 0; // A single number to store the time when we should deactive the current night light cycle
bool nightLightOn = false; // enable re-activation even when already activated

int LDR = 0;
int LDRValue = 0;
int Light_Sensitivity = 500;

int led = 13;               // the pin that the LED is atteched to
int sensor = 2;             // the pin that the sensor is atteched to
int state = LOW;            // by default, no motion detected
int val = 0;                // variable to store the sensor status (value)


void setup() {
  pinMode(nightLight,OUTPUT);
  pinMode(trigger,INPUT);
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);

  Serial.begin(9600);

  pinMode(led, OUTPUT);      // initalize LED as an output
  pinMode(sensor, INPUT);    // initialize sensor as an input

  analogWrite(RED,0);
  analogWrite(GREEN,0);
  analogWrite(BLUE,0);

  // minutes until begin sunrise
  //minutesUntilSunrise = hoursUntilSunrise * 60;
  minutesUntilSunrise = 1;
  Serial.print("Minutes until sunrise:");
  Serial.println(minutesUntilSunrise);
}

void loop(){

  clock();

  sunrisealarm();

  motionTriggered();

  delay(100);

}

void clock(){
  if(millis() >= previousMillis+86400000){
    // a full day has elapsed, reset the clock;
    previousMillis +=86400000;
  }
  currentMillis = millis() - previousMillis; // this keeps our currentMillis the same each day
  currentMinutes = (currentMillis/1000)/60;
  Serial.println(currentMinutes);
}
```

Dineth Hettiarachchi
CSE 3323

```
void sunrisealarm(){
   //each second during the 30 minite period should increase the colour value by:
  float increment = (float) 255/(1*60);

  if(currentMinutes >= minutesUntilSunrise){
     //sunrise begins!
     float currentVal = (float)((currentMillis/1000) - (minutesUntilSunrise*60)) * increment;
     Serial.print("Current value for sunrise:");
     Serial.println(currentVal);
     //during ramp up, write the current value of minutes X brightness increment
     if(currentVal < 255){

         analogWrite(RED,currentVal);
         analogWrite(GREEN,currentVal);
         analogWrite(BLUE,currentVal);

     }
     else if(currentMinutes - minutesUntilSunrise < 4){
         analogWrite(RED,255);
         analogWrite(GREEN,255);
         analogWrite(BLUE,255);
     }
     else{
        analogWrite(RED,0);
        analogWrite(GREEN,0);
        analogWrite(BLUE,0);
      }
  }
}

void nightLamp () {
  LDRValue = analogRead(LDR);
  Serial.println(LDRValue);
  if(currentMinutes < minutesUntilSunrise){
    if (LDRValue > Light_Sensitivity) {
        analogWrite(BLUE,1);
    } else {
      analogWrite(BLUE,0);
    }
  } else {
      //analogWrite(BLUE,0);
  }
}

void motionTriggered(){
  val = digitalRead(sensor);   // read sensor value
  if(currentMinutes < minutesUntilSunrise){
    if (val == HIGH) {           // check if the sensor is HIGH
      if (analogRead(LDR) > Light_Sensitivity) {
        analogWrite(RED,255);
        analogWrite(GREEN,255);
        analogWrite(BLUE,0);
      }
      if (state == LOW) {
        Serial.println("Motion detected!");
        state = HIGH;       // update variable state to HIGH
      }
    }
    else {
        analogWrite(RED,0);
        analogWrite(GREEN,0);
        analogWrite(BLUE,0);
        nightLamp ();
        if (state == HIGH){
          Serial.println("Motion stopped!");
          state = LOW;       // update variable state to LOW

      }
    }
  }
}
```