

**Design with Microprocessors Project Documentation**

**“Night Lamp”**

**Antonescu Maria-Cristina**

**Group 30431**

## Structure of documentation

1. Introduction.....	pg3
2. Proposed solution and implementation.....	pg3
3. Testing and validation.....	pg5
4. Conclusion.....	pg6

## 1. Introduction

I have chosen to implement for this project an app-controlled night-lamp, because I thought it would be very useful for my personal life. I actually use night lamps quite often, since I am unable to sleep in the dark. I have recently purchased a lightbulb which can change the color of the light when you press a button, so this is the main inspiration for this project.

## 2. Proposed solution and implementation

### 2.1. Description of proposed solution

I have decided to develop a circuit which enables the user to connect through Wi-Fi and choose the desired color of the night lamp. There are 7 colors to choose from, which I consider to be a good variety. After the user has selected their desired color, they can press the "START" button, which will start the countdown until the lamp is automatically turned off. If the lamp is off and there is motion detected, then the lamp will turn on and will stay on for a few seconds.

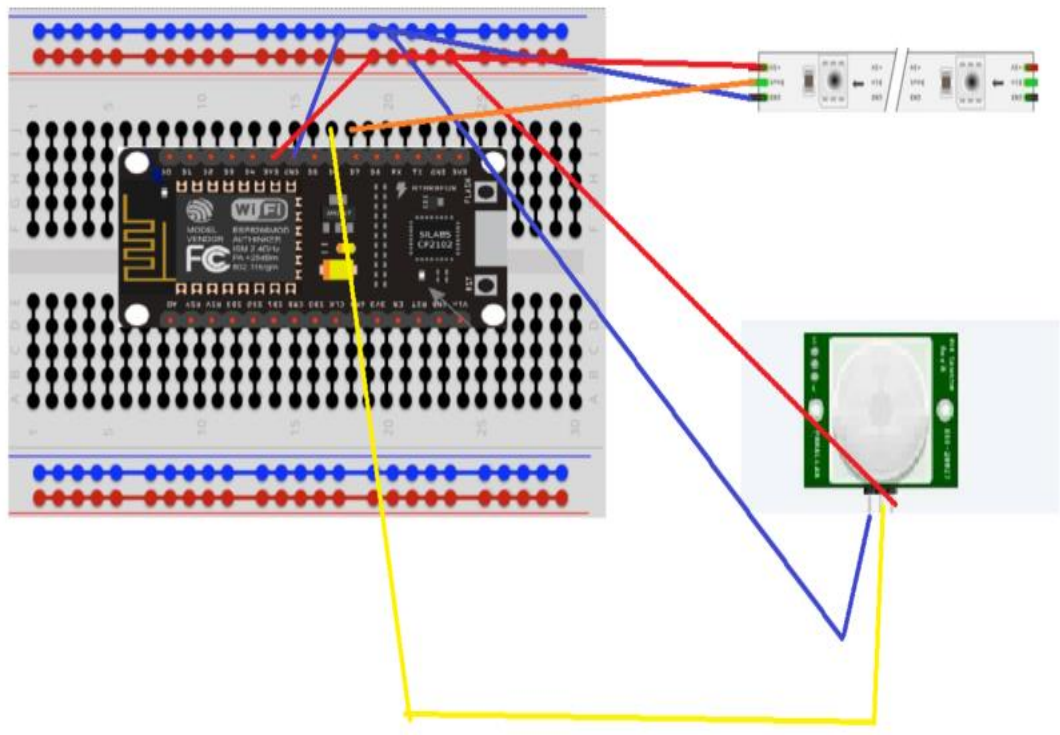
The development board will connect to the Wi-Fi and automatically create a webpage which has 8 buttons: 7 colors and one START button which will start the 10 second countdown until the LEDs turn off. Each button sends a HTTP request to the board, which contains the name of the button the user has pressed. According to this, the LEDs output pin will receive the color as information. The program stores the state of the LEDs in a Boolean variable, so that the commands can be processed accordingly and the state will be updated.

### 2.2. Hardware components

The circuit consists of 3 components:

1. ESP 8266 development board
2. RGB LEDs wheel
3. PIR motion sensor

The schematic of the circuit is the following:



## 2.3. Software part

I used in total 5 libraries:

```
<ESP8266WiFi.h>
<WiFiClient.h>
<ESP8266WebServer.h>
<Adafruit_NeoPixel.h>
<avr/power.h>.
```

The first three are used for the Wi-Fi connection, and the next two are used for the LED wheel.

For turning on the LEDs, I used the function from the strandwheel-test, `colorWipe()`, which had as parameters the RGB codes of the desired color.

For setting up the Wi-Fi connection and for managing the HTTP requests, I used built-in functions, such as `server.send()` and `server.on()`. `Server.on()` made working with HTTP request much easier, since the syntax proved to be quite uncomplicated. `Server.send()` sends the HTML page to the client, which was stored in a string.

```

webpage = "<!DOCTYPE html><html><body><h1>Welcome to our Night
Lamp App!</h1> <h2> Please Select your preferences:</h2><p><a
href=\"/RED\"><button      class=\"button\">RED</button></a></p><p><a
href=\"/GREEN\"><button    class=\"button\">GREEN</button></a></p><p><a
href=\"/BLUE\"><button      class=\"button\">BLUE</button></a></p><p><a
href=\"/CYAN\"><button      class=\"button\">CYAN</button></a></p><p><a
href=\"/MAGENTA\"><button
class=\"button\">MAGENTA</button></a></p><p><a
href=\"/YELLOW\"><button  class=\"button\">YELLOW</button></a></p><p><a
href=\"/WHITE\"><button    class=\"button\">WHITE</button></a></p><p><a
href=\"/START\"><button
class=\"button\">START</button></a></p></body></html>";

```

The HTML page is a very basic one, since styling the elements made them not appear at all.

The delay() function has been used to make the LEDs turn off after 10 seconds and to make a delay between processing the requests.

### 3. Testing and validation

My first idea for this project was to create an app using MIT App Inventor, which enables the user to select there their preferences. The board was supposed to connect to my hotspot and then the app was supposed to generate HTTP requests through which the color selected by the user was transmitted. Although the board managed to connect to my hotspot, when I tried to scan the QR code of the app, it failed to connect each time I tried.

The second approach was to generate a webpage like in the laboratory guide, which had all the buttons necessary to select the preferences. I think the most difficult part of this project was figuring out how to send the data to the board through HTTP requests. The second approach still didn't function properly.

The third and final approach was using built-in functions which handle HTTP requests. This proved to be the best solution. I first tested a simple LED-ON/LED-OFF app with my LED wheel, which turned out to work. Then I added all the buttons for the colors. Then I tested each one to see if the LED will turn on with the desired color. The last step was configuring the START menu, which starts the countdown. Here I encountered a few problems as well, since I tried using interrupts and the millis() function, but it ended up working only with delay(). The PIR motion sensor has a few problems as well, since it is

too sensitive to movement or it doesn't detect movement at all. I haven't managed to find out a solution for that problem yet.

Then I tested all the colors with the START button, to make sure that the LEDs will turn off after 10 seconds.

#### 4. Conclusion

Overall, I think the basic functionalities have been fulfilled, however there are many improvements to be made. The way the lamp is controlled can be through an app with a much more pleasant design. Through that app, the user can also select preferences regarding time after which the lamp should be shut off. I also think that these events can be processed using interrupts, to assure the good performance of the app. The motion sensor can also be improved regarding its sensitivity to motion. Furthermore, it can compare the values read over a period of time to ensure that the lamp doesn't turn on unless there is actual movement in its proximity.

This has been a fun project to work with and I really enjoyed the way I needed to combine both my hardware and my software knowledge for this. The part that I enjoyed the most was sending data to the board, since I wasn't very familiar with how HTTP requests worked before.