Arduino wakeup light documentation

By Yubo Qiu

August 9, 2015

This functional specification will be available in PDF format for those who don’t have/use Word.

Change Log

Author	Date	Notes
Yubo Qiu	Aug 9, 2015	Initial Draft

Table of Contents

1. Introduction

1.1 Brief History of this project

This project started because my mother said that she was always waking up tired in the winter. After some research, I figured out that she was tired because the alarm clock would jolt her awake, instead of being woken up by the sunrise, like in summer. A quick search suggested that the Philips wake-up light would be a great solution to this, but costed $100 or so. I wanted a cheaper solution, so I decided to go the DIY route. I figured that since I already had some background in PCB development, this would be a great opportunity to go from initial concept to final delivery. I can also learn a bit about microcontrollers along the way!

Of course, the first thing was to look online, but I quickly found that those DIY solutions were unpolished. Of course, my solution might end up the same, but at least I'll have control over that and decide the tradeoffs on my own terms. So, I ended up deciding to build this entire project from scratch, just to know how it feels.

1.2 Purpose of this document

This document is intended to describe functional specifications of the wake-up light. This document specifies the all components of this project, including PCB, firmware, mechanical and debugging. The intended audience is for anyone looking for a practical beginner project, and for developers to make sure that all design criteria are met.

1.3 Scope of the project

The wakeup light gradually increases the brightness of the LED strips to simulate a sunrise. This provides a more natural wake-up, rather than an alarm clock to jolt you awake. This project is also intended to be a starting point for anyone else wanting to build a wake-up light themselves—it doesn't require advanced knowledge of circuit design or programming, instead, it requires you to follow through.

2. General Description

The wakeup light gradually increases the brightness of the LED strips to simulate a sunrise. Since all the code is on Github, I encourage the end user to change the code to make the light better.

2.1 Functions

2.2 Cost

The cost is at around $30 for hardware alone, however, depending on where you source components, you may get it cheaper. You do not need quality components, stuff from eBay is fine (that's where I sourced my stuff).

3. Functional Requirements

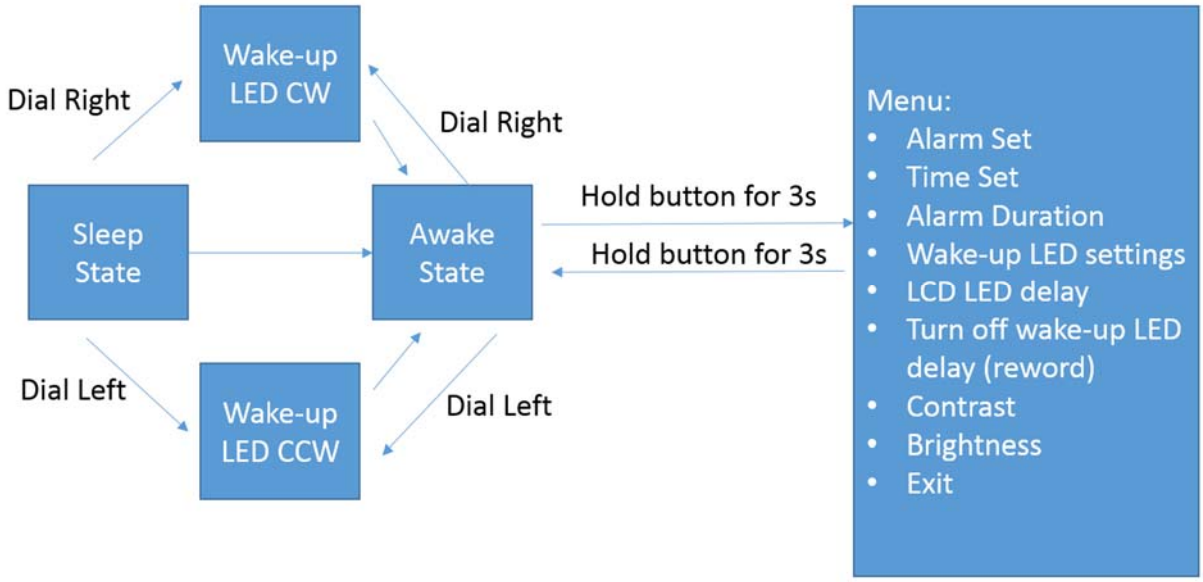
3.1 User Interface

The rotary encoder controls the majority of the configurations, where turning the encoder scrolls through the menu items, and pushing the button selects the item. such as alarm/clock/date set, alarm on/off, and light on/off. When the alarm is set, the Arduino gradually turns up an LED strip over a specified period (default 30min). Another functions include a delayed off, where you gradually dim the light before turning off.

3.2 Software

Stuff happens here

3.2.1 State machine for rotary encoder



3.2.1.1 Sleep State

By default the clock is in sleep state, where the LCD LED is off, and the clock and date is shown

3.2.1.2 Awake State

When the dial is turned left/right or pressed, then the clock will enter awake state where the LCD LED will turn on.

3.2.1.3 Menu state

The menu state will be displayed when the button is held for 3s. TO leave the menu state, hold the button for 3s, or navigate to exit.

3.2.2 Wi-Fi

Add wi-fi compatability to control from web-browser

3.2.3 Mobile app

Android or iOS app that connects to the device through wifi

3.3 Block diagram

3.4 Hardware

- 16x2 LCD with backlight
- Custom PCB
- Arduino nano or mini
- External Clock oscillator
- Push button
- Caps and resistors
- LED light strip

4. Mechanical

- All components must fit in a LxWxH: 3.5” x 3.5” x 1.5” enclosure
- Rotary encoder will protrude from the middle of the enclosure

5. Debugging

6. Further development

Wi-fi integration for android/iOS
Remote control functionality
Add a python wrapper