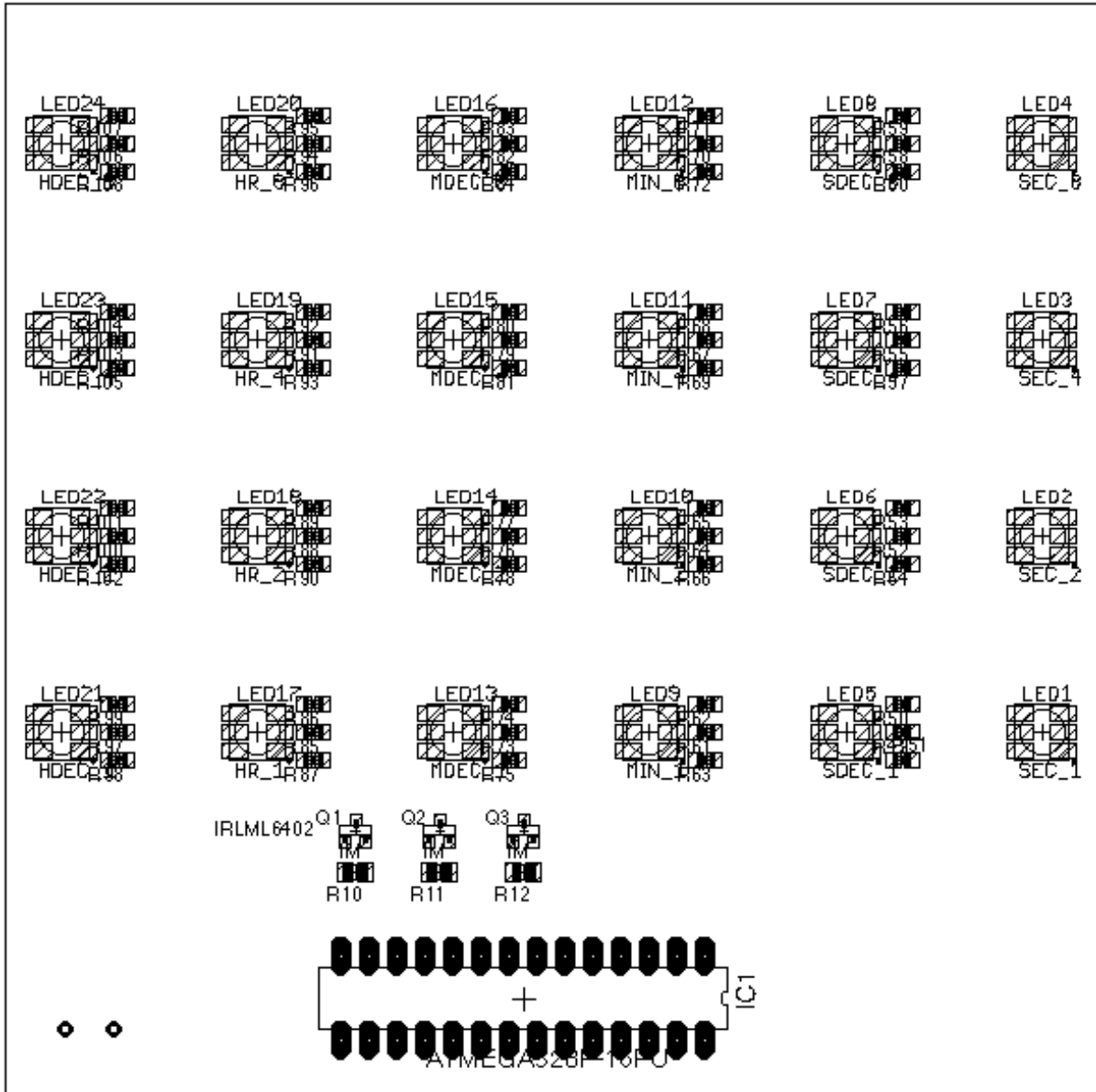


# NERDCLOCK.

An Open Source Arduino® RGB Binary Clock



## User Manual

Version 1.0 - English

Welcome to your new NerdClock Open Source Arduino® RGB Binary Clock.

Thank you for being a part of the project, thank you for being awesome :D

With this user manual you will learn to know use all features of the the NerdClock!

## Table of Contents:

### Chapter 1 - Introduction

1.1 Clock Safety Notice. . . . .	3
1.2 Clock Specification. . . . .	3
1.3 Clock Features . . . . .	3

### Chapter 2 - Getting Started

2.1 Getting Started: Power On/Off. . . . .	4
2.2 Getting Started: Time Reading. . . . .	5
2.3 Getting Started: Time Setup. . . . .	6
2.4 Getting Started: Color Setup . . . . .	6

### Chapter 3 - Additional Information

3.1 Firmware Reflash/Update. . . . .	7
3.2 Custom Non-USB Power Supply. . . . .	8
3.3 Expanding NerdClock w/ DS1307 RTC Module . . . . .	8

### Chapter 4 - Appendix

4.1 Circuit Diagram: Control Circuit . . . . .	9
4.1 Circuit Diagram: Cathode Circuit . . . . .	10
4.1 Circuit Diagram: LED Circuit . . . . .	10

## 1.1 - Clock Safety Notice



NerdClock contains highly electrostatic sensitive MOSFETs and integrated circuits. Before touching the assembled clock, please make sure that you are statically discharged.

The NerdClock is designed for indoor use only.

## 1.2 - Clock Specification:

- Operation voltage range: Regulated 3.3 V - 5.2 V DC
- Maximum current draw:  $\leq 750$  mA @5.0 V DC (Assembly Checker, all 24 LEDs on, white)
- Maximum power dissipation:  $\leq 4$  W, 1 W typically
- PCB dimension: 99 mm (L) x 99 mm (W) x 1.2 mm (H)
- Dimension w/ components: 99 mm (L) x 99 mm (W) x 22,5 mm (H)
- PCB weight: 25 g
- Weight w/ components: 37 g
- Default microcontroller: Atmel® AVR® Atmega328P-16PU with Arduino® Bootloader
- RGB LED Type: 5050 (Vf\_Red: 1.8 V; Vf\_Green: 2.8 V; Vf\_Blue: 2.8 V)

## 1.3 - Clock Features:

- 24 hour binary coded decimal clock mode
- Static 8-bit latched shift register driving, 24x individually controllable 6x4 RGB LED matrix
- RGB spectrum fade
- R, G, B manually adjustable using trimm potentiometer



## 2.3 - Getting Started: Time Reading

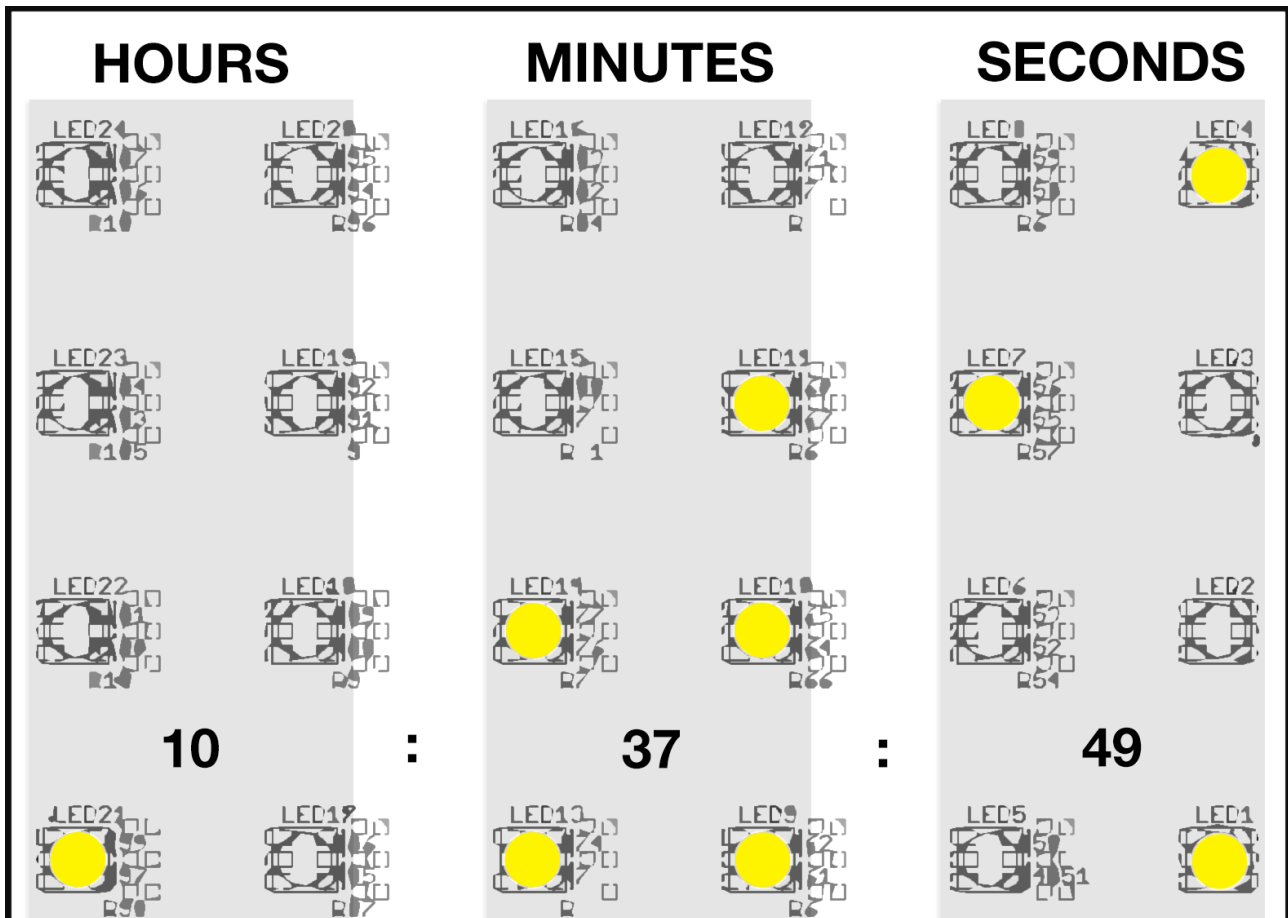


Figure 2.3.1

We are used to reading *decimal* numbers (based on number 10). Binary clocks however use *binary* numbers (based on number 2) with either '1' or '0' to code decimal representations while an LED on represents '1'. *NerdClock* is also known as a *BCD* (**B**inary **C**oded **D**ecimal) clock. To understand the time, a conversion from *BCD* to *decimal* numbers is required.

*Figure 2.3.1* shows *NerdClock*. For better visualization, the clock is divided into 3 sections vertically which are then divided into two columns each with the leftmost two columns representing the 1st and the 2nd digit of the hour. Each horizontal row represents a power of two. The bottom one shows  $2^0 = 1$ , the topmost  $2^3 = 8$ .

- Begin reading the first hour digit. LED for 2 is off and the LED for 1 is on. That means  $0 \times 2^1$  and  $1 \times 2^0 \rightarrow 0 + 1 = 1$ .
- Continue reading the second hour digit. All LEDs are switched off.  
 $0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 0 + 0 + 0 + 0 = 0 \rightarrow$  It's 10 AM.
- Repeat for minutes:  $2 + 1 = 3$  and  $4 + 2 + 1 = 7 \rightarrow 10:37$  AM.
- Complete reading: 10:37:49

**Further reading (internet connection required):**

<http://www.wikihow.com/Read-a-Binary-Clock>

<http://www.exploringbinary.com/how-to-read-a-binary-clock/>

## 2.4 - Getting Started: Time Setup

NerdClock comes with 6 different buttons with 5 that are used for time setting (Figure 5.1).

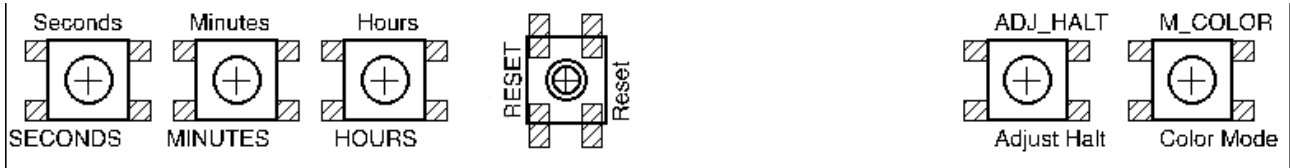


Figure 5.1

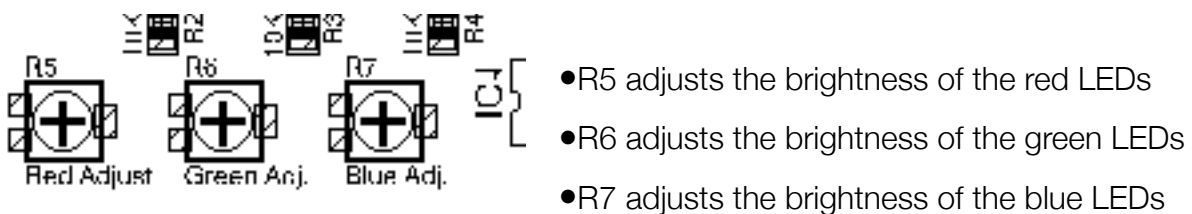
<b>Seconds:</b>	<b>Minutes:</b>	<b>Hours:</b>	<b>Reset:</b>	<b>Adjust Halt:</b>
Increases the seconds by one.	Increases the minutes by one.	Increases the hours by one.	Resets the clock to 00:00. Equivalent to power off + power on.	Stops the clock. Press button again to resume.

## 2.5 - Getting Started: Color Setup

By default, NerdClock starts fading through a predefined RGB color cycle as soon as it turns on. The predefined color cycle transitions between the following colors:

Blue ► Green ► Red ► White ► Orange ► Light Blue ► Purple

To set the color manually, press the Color Mode button. NerdClock will now read the values provided by the three potentiometers.



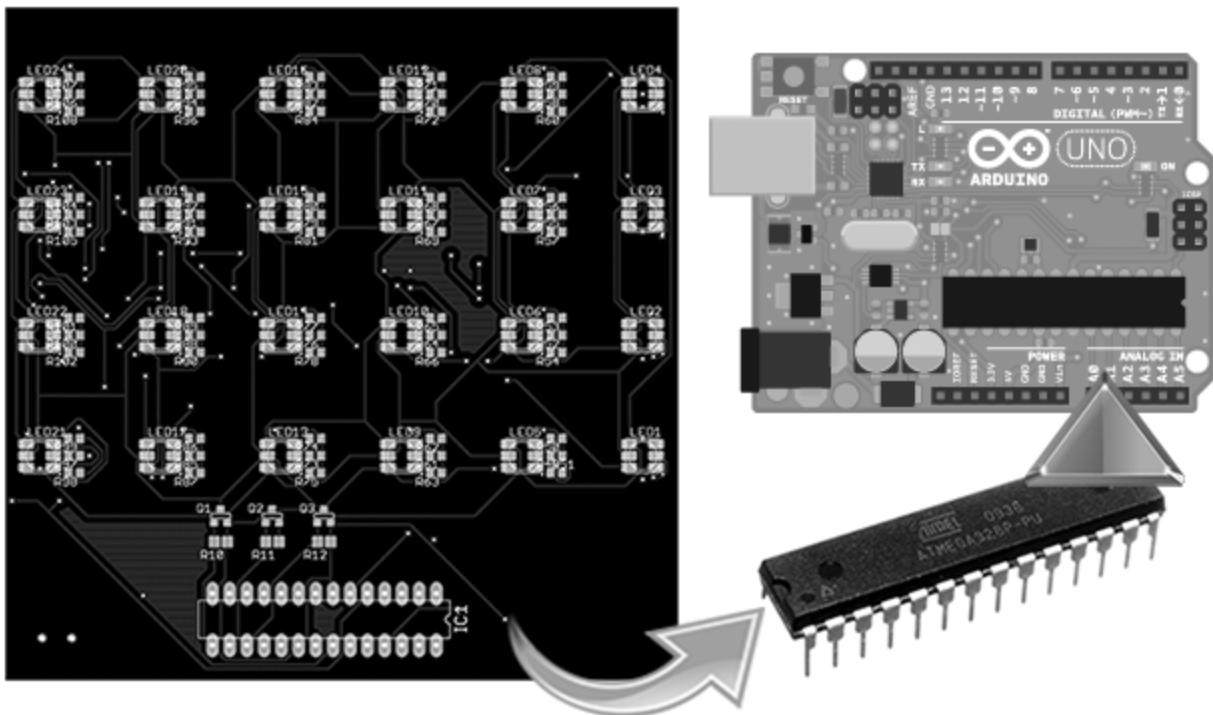
Twisting the potentiometers in clockwise direction to increase the brightness value; counterclockwise to decrease.

### 3.1 - Firmware Reflash/Update

Check for NerdClock firmware updates: <http://cpumuseum.jimdo.com/store/nerdclock/>

NerdClock is completely written in Arduino® C language and utilizes a standalone Arduino® platform. This is to show you how to reflash or update the NerdClock firmware.

- **Step 1:** Remove the Atmel® AVR® Atmega328P-16PU from NerdClock using a IC remover or lift up the microcontroller with a flathead screwdriver.
- **Step 2:** Insert the microcontroller into the microcontroller socket of the Arduino® Uno.



- **Step 3:** Connect the Arduino® Uno to a computer.
- **Step 4:** Download, install and start the Arduino® IDE (<https://www.arduino.cc/en/Main/Software>), requires Java (<https://java.com/download/>)
- **Step 5:** Open the new firmware (File -> Open). The firmware has the filetype \*.ino

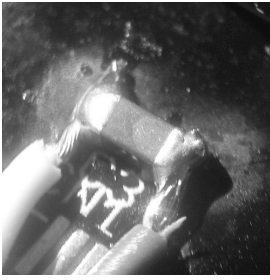


- **Step 6:** Upload the new firmware by clicking on the arrow.
- **Step 7:** Remove the connection between the Arduino®

Uno and the computer.

- **Step 8:** Remove the Atmel® AVR® Atmega328P-16PU from the Arduino® Uno using a IC remover or lift up the microcontroller with a flathead screwdriver.
- **Step 9:** Insert the microcontroller into the microcontroller socket of the NerdClock.

## 3.2 - Custom Non-USB Power Supply



In addition to the USB power supply port, other points of the PCB can be used to supply the 5V voltage. **This will necessarily happen if the clock will stand on a platform and not get mounted to a wall.**

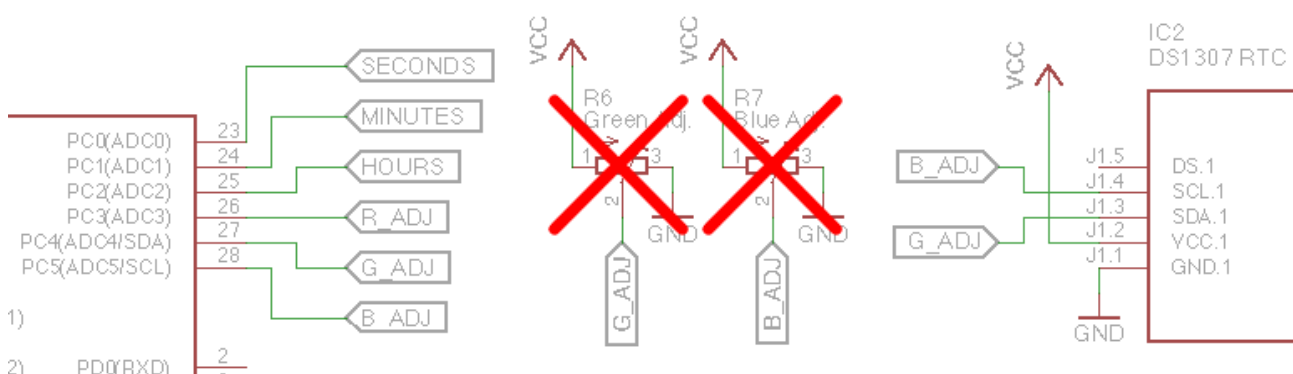
Figure 3.2.1

It is recommended to use the two pads of C3 to solder non USB power supply wires as shown in figure 3.2.1. The darker, right wire is connected to positive 5 V; the left wire is connected to GND. The power supply must supply 5 V (at least 3.3 V) regulated DC. This can come from a step down converter, a 5 V wall wart or a battery boosted to 5 V.

## 3.3 - Expanding NerdClock w/ RTC Module

By default, NerdClock does not have a RTC. It means that the time is lost when disconnecting NerdClock from the power source. In order to keep the time running even if the clock is off, it is possible to add an RTC module.

However, installing a RTC module will only work at the expense of the manual color adjust potentiometers: The green and blue potentiometers (R6, R7) are connected to the I<sup>2</sup>C protocol pins of the AVR<sup>®</sup> microcontroller. Therefore a RTC module can only be installed when desoldering the potentiometers. The following excerpt of the NerdClock circuit diagram demonstrates how to connect a DS1307 RTC module to NerdClock.



Desolder the crossed out potentiometers R6 and R7, connect pin 27 of the AVR<sup>®</sup> microcontroller to SDA of the DS1307 RTC module and pin 28 to SCL. VCC is 5 V.



# 4.1 - Circuit Diagram: Control, Cathode, LED Circuit

