# Table of Contents

## Overview  
5

## Choosing Your Components  
7

## Construction and Wiring  
12
- Wiring the Doubler Board  
- Assembling the Neo-Pixel Ring  

## Software Installation  
18

## Installing Libraries  
18

## Sample Code and Audio Files  
20

## Operating the Clock  
22
- Infrared Controls  
- BLE Connect  
- Touch Controls  
- Serial Monitor Debugging Controls  
- Setting the Time  

## The Setup Menu  
30
- Audio Mode  
- Brightness Mode  
- Daytime Begins and Nighttime Begins  
- Daytime Volume Level  
- Nighttime Volume Level  
- Daytime Brightness Level  
- Nighttime Brightness Level  
- Chimes Option  
- Voice Announcement  
- Hourly Audio  
- Quarterly Audio  
- Animation Mode  
- Music Mode  
- Alarm Mode  

## 3D Printed Case  
34

## Assembling the Clock  
36
- Mount the Speakers  
- Mount the Doubler Board  
- Touch Controls  
- NeoPixel Ring  
- Frame Ring Assembly  

## Modifying the Models  
40
- About Fusion 360  
- Loading the Model  
- User Defined Parameters  
- Split versus Solid
• Filling the Holes
• Export to STL

Advanced Configuration

• Defining the Top Pixel
• Calibrating Touch Controls
• Initial Defaults
• Changing IR Codes
• Calibrating the Photocell
Overview

When Adafruit first introduced the 60 element Neo-Pixel ring in March 2004, the obvious choice for a project was to build a clock. By selectively lighting certain pixels around the ring you can depict analog time in a digital manner.

NeoPixel 1/4 60 Ring - 5050 RGB LED w/ Integrated Drivers
The biggest NeoPixel Ring yet! With four of these you can make a huge ring with 60 ultra bright smart LED NeoPixels are arranged in a circle with a 6.2" diameter. Each...
https://www.adafruit.com/product/1768

Because the SAMD21 processor used in a Feather M0 boards has a built-in real-time clock, it's easy to keep track of time once you start the clock. The "RTC_Zero" library provided by Arduino.cc makes it easy to set and read the time using this built-in real-time clock hardware.
Adafruit Feather M0 Bluefruit LE
Feather is the new development board from Adafruit, and like its namesake, it is thin, light, and lets you fly! We designed Feather to be a new standard for portable microcontroller...

https://www.adafruit.com/product/2995

We give you a variety of ways to set the time and control various options on the clock. The simplest method is a set of 4 touch control pads using the Adafruit_FreeTouch() library. But if you don't want to bother to take the device down off of the wall, we've also included the capability of using an infrared remote, or you can connect to the device using the Adafruit Bluefruit Connect app available for iOS() or Android().

Once we designed the clock, we decided it would be nice to have it chime and/or play music every hour or quarter hour. You can optionally add the Music Maker Feather Wing with 3W amplifier and a small pair of speakers. Not only does this give you the ability to play chimes and/or music, by recording a small set of spoken phrases, the clock can also speak the current time. Furthermore we have implemented a voice prompt menu system for setting a variety of options. This project serves as a demonstration of how you could add a voice prompt menu system to any project where you have to set options but do not have any sort of text display available.

The MP3 files for the voice output and music output are stored on an SD card on the Music Maker Wing. This tutorial and code has been updated to allow you to drag-and-drop MP3 files onto the device by connecting it to your PC where it will appear as a USB flash drive.

Adafruit "Music Maker" MP3 Shield for Arduino w/3W Stereo Amp
Bend all audio files to your will with the Adafruit Music Maker shield for Arduino! This powerful shield features the VS1053, an encoding/decoding (codec) chip that can decode a wide...

https://www.adafruit.com/product/1788
We've added a series of animated color patterns that can be displayed at the top of the hour. You can set an alarm to go off at a particular time. You can play any MP3 music file at the top of the hour. We also give you the option to set up nighttime hours so that the volume level is lower and the display is dimmed at night. There is even option to add a photocell that will lower the volume and brightness if the room is dark rather than using a time-based volume and brightness level reduction.

The code written in C++ for the Arduino IDE is set up with a series of conditional compile flags that are easy to change so that you can turn off or on any of the features we have described. So if you're going to use Bluetooth but not infrared or touch controllers and don't have a photocell sensor, it's easy to simply set a flag or two and recompile the code for just options you've chosen to use in your project.

We have included a design for a 3D printed faceplate and frame that includes a wall hanger or an optional tabletop stand. Not only have we included STL files, we also include a Fusion 360 model so that you can adapt the design to your own liking. Or you are free to build any kind of faceplate or case of your own design using any other materials. Just mount the ring and electronics in it as you like.

Choosing Your Components

The 60 element NeoPixel ring is sold as a quarter segment of 15 pixels each so you will need a total of four of these to complete the full ring.

The code we provide depends upon having a SAMD21 processor to make use of its built-in real-time clock. Although any of the Adafruit boards which use the SAMD21 processor could be used, we have chosen to use the Feather form factor so that we have the option to connect it to the Music Maker Feather Wing. If you're going to use the option of BLE control to set the time and control various features of the clock then you will have to use the Feather M0 Bluefruit LE.

Adafruit Feather M0 Bluefruit LE
Feather is the new development board from Adafruit, and like its namesake, it is thin, light, and lets you fly! We designed Feather to be a new standard for portable microcontroller...
https://www.adafruit.com/product/2995
If you are not using Bluetooth control then either the Feather M0 Basic or Feather M0 Express can be used. These boards are slightly less expensive if you don't plan to use the Bluetooth feature.

**Adafruit Feather M0 Basic Proto - ATSAMD21 Cortex M0**

Feather is the new development board from Adafruit, and like its namesake it is thin, light, and lets you fly! We designed Feather to be a new standard for portable microcontroller...

https://www.adafruit.com/product/2772

**Adafruit Feather M0 Express**

At the Feather M0's heart is an ATSAMD21G18 ARM Cortex M0+ processor, clocked at 48 MHz and at 3.3V logic, the same one used in the new

https://www.adafruit.com/product/3403

Note if you are using the M0 Express, we will still be programming it using the Arduino IDE. We have not implemented any code for this project using CircuitPython.

If you’re going to use the music, chimes, and/or voice menus then you need the Music Maker Feather Wing and a speaker or two.

**Music Maker FeatherWing w/ Amp - MP3 OGG WAV MIDI Synth Player**

Bend all audio files to your will with the Adafruit Music Maker FeatherWing + Amplifier! It's a fun-size version of our

https://www.adafruit.com/product/3436
Although the image above shows speakers, they are sold separately. We recommend this pair of enclosed stereo speakers.

**Stereo Enclosed Speaker Set - 3W 4 Ohm**

Listen up! This set of two 2.8” x 1.2” speakers are the perfect addition to any audio project where you need 4 ohm impedance and 3W or less of power. We particularly like...


You will need a micro SD memory card to store the MP3 files containing the voice prompts, chimes, and music files. The sample files we provide required just over 25 MB so this 8 GB card or anything similar you might already have on hand is way more than sufficient.

**SD/MicroSD Memory Card (8 GB SDHC)**

Add mega-storage in a jiffy using this 8 GB class 4 micro-SD card. It comes with a SD adapter so you can use it with any of our shields or adapters. Preformatted to FAT so it works out...

[https://www.adafruit.com/product/1294](https://www.adafruit.com/product/1294)

You can connect your Music Maker Wing to your Feather M0 using female headers but we decided to use the Feather Wing Doubler Prototype board to mount them side-by-side and to give us some prototyping space for the IR receiver and the photocell options.
FeatherWing Doubler - Prototyping Add-on For All Feather Boards
This is the FeatherWing Doubler - a prototyping add-on and more for all Feather boards. This is similar to our https://www.adafruit.com/product/2890

The design for the 3D printed case in this guide assumes you are using the Doubler and the set of speakers shown above. You can feel free to adapt the design however you want and connect the components to meet your needs.

This project is powered through the USB connector of the Feather M0 and it must be plugged in at all times. The NeoPixels operate off of the 5 volts from the USB. So you will need a good USB power supply. If you do not turn up the brightness on the NeoPixels very high or do not use animations that light up all of the pixels it can reduce your power requirements. We have had success with this supply.

5V 2A Switching Power Supply w/ USB-A Connector
Our 5V 2A USB power adapter is the perfect choice for powering single-board computers like Raspberry Pi, BeagleBone, or anything else that's power-hungry! This adapter was... https://www.adafruit.com/product/1994

If you are going to use an infrared remote to set the time and control the features of the clock you will need an IR receiver such as this one.
IR (Infrared) Receiver Sensor
IR sensor tuned to 38KHz, perfect for receiving commands from a TV remote control. Runs at 3V to 5V so it's great for any microcontroller. To use, connect pin 3 (all the...
https://www.adafruit.com/product/157

You can use a wide variety of remote controls for TV, DVD and other consumer electronic devices as long as they use a protocol supported by IRLib2. It is an Arduino compatible library for receiving, decoding and transmitting IR signals. We have preconfigured the software to use the Adafruit Mini Remote shown here that uses NEC protocol. So if you already have one of these remotes or purchase one you will definitely be ready to go. If you choose to use your own remote, we will tell you how to configure it in the "Advanced Configuration" section.

Mini Remote Control
This little remote control would be handy for controlling a robot or other project from across the room. It has 21 buttons and a layout we thought was handy: directional buttons and...
https://www.adafruit.com/product/389

You can also optionally add a photocell so that the volume level of the speakers and/or the brightness of the NeoPixels is reduced at night when the ambient light in the room is below a certain threshold which you can set. If you use the photocell, you also need a 10K resistor.
Construction and Wiring

Wiring the Doubler Board

You will need to solder 3 wires from the doubler board that will later attach to the NeoPixel ring. We recommend red, green, yellow wires. Make them about 6 inches long and you can trim them off shorter once you do the final assembly.

We recommend you solder the wires and other components to the doubler before soldering the female headers for the Feather M0 and Music Maker Wing. We soldered the headers first and they got in the way of soldering the other components. We accidentally touched the side of the soldering iron to the header and damaged it. So learn from our mistakes.

The image below shows the doubler board from the bottom side. The green wire connects to the ground bus anywhere along that column of holes. The yellow data wire connects to pin 11. The red power wire should connect to the USB pin to obtain +5 volts.
When prototyping this designed we used the USB pin as shown here. But when wiring up the final version of the project, we mistakenly connected the red power wire to the 3.3v bus. That was a mistake but it worked anyway. You can drive NeoPixels at 3.3v however it will put a strain on the 3.3v regulator of the Feather if you turn up the brightness very high. So even though it worked, we still recommend you use the USB pin as shown here.

If you're using the IR receiver and/or the photocell, these components will also be soldered to the bottom side of the doubler board.

Insert the IR receiver so that it's first pin goes in the hole that connects to pin 12 as shown in the image below. Bend the pins 90° so the component lies flat and the lens of the receiver points away from the board. You also need to solder in a power and ground wire to supply that device. Bridge these wires to the center and outer pins of the device. Note that the power pin MUST connect to 3.3v and NOT USB 5v.
If you are using the photocell option, you will need to solder in a 10K resistor in addition to the photocell. As shown in the above image, connect one end of the resistor to the ground bus. Connect the other end to pin A4. Also connect one end of the photocell to A4 by putting it in the adjacent hole and bridging them together on the opposite side of the board. The other end of the photocell goes to the 3.3v bus.

If you are using the touch control option, solder 4 wires to pins A0 through A3. Although the IR sensor and photocell had to be on the bottom side of the board, these control wires can go on either side of the board. We put them on the top side rather than the bottom where the other components are. This is because we had already soldered in the female headers and it would be difficult to solder the wires in place. They can be any color wires and you should make them about 6 inches long. You can trim them off to appropriate length later.

The image below shows the location of the touch control wires. It also highlights the places where you need to bridge together adjacent pins to properly connect the IR receiver to its power and ground wires as well as the connection between the 10K resistor and the photocell on pin A4.
After all of the components and connecting wires have been soldered in place, then solder in 2 sets of female headers. Then solder a set of male header pins to the Feather M0 board and the Music Maker Feather Wing. The Music Wing comes with a 4 screw terminal block. You have the option of installing it or soldering the speaker wires directly to the wing. We recommend using the screw terminals in case you ever need to take the device apart for some reason.

Temporarily insert the Feather M0 and Music Wing into the doubler board. You may need to remove them later to mount the doubler board onto the back of the 3D printed faceplate.
Assembling the Neo-Pixel Ring

The 60 pixel ring comes in 4 segments of 15 pixels each. We used superglue to connect the 4 segments together. You then need to solder together the power, ground, and data pads of 3 out of the 4 joints. The joint that is not soldered will be at the top of the clock.

As you are looking at the backside of the ring, the red, green, and yellow wires from the clock module will connect to the ring to the left of the joint as shown below. We temporarily soldered the 3 wires from the clock module to the ring before mounting everything into the 3D printed case. That way we were able to test all of the components before final assembly.
If you're using the speakers that we have recommended, you'll need to cut off the jack from the speaker wires. Strip the ends of the wires, twists the strands and we recommend that you tin the wire tips with solder. That will make it easier to put them into the screw terminals of the Music Maker Wing. The red wires go in the positive terminals which are the two outside terminals. The black wires go on the two inner terminals. We left the wires quite long while testing before assembling it in the 3D printed case. You can always cut them off shorter and re-tin them after final assembly.

You should also strip and tin the loose ends of the 4 touch control wires. We will eventually wrap each one around a screw and/or solder each wire to a screw.
Once you have connected everything, go to the next section which describes how to install the software and the audio files.

Software Installation

If you are unfamiliar with the Arduino IDE and/or have never uploaded programs to a Feather M0 you should familiarize yourself with the following section of the Feather M0 BLE Learning Guide. The instructions on these links will show you how to configure your Arduino IDE for use with Adafruit boards and libraries.

- Arduino IDE Setup ()
- Using with Arduino IDE ()
- Installing BLE Library ()

We recommend that you run some of the BLE sample sketches to confirm that it is working properly. You might also want to take the opportunity to update your BLE firmware according to the instruction in those guides.

If you're not using the BLE option you need not install the BLE library. There are other libraries that you will need as described on the next page. You also have to download the software sketch for this project along with a collection of MP3 files. That is covered in the following sections.

Installing Libraries

You will need the "RTCZero" library available from Arduino.cc. You can install it from the Arduino IDE. In the Arduino IDE, click on the menu "Sketch->Include Library->Manage Libraries". Then in the search box type "RTCZero". Hover your mouse over the lower right corner and click on "Install".
You also need to search for and install the "Adafruit NeoPixel", "Adafruit DMA NeoPixel", and "Adafruit FreeTouch" libraries using the same procedure to search for and install them.

If you want to be able to drag and drop your MP3 files onto the board you will also need the "Adafruit TinyUSB Library" which is not available via the library manager as of this writing. You will have to download it directly from GitHub at https://github.com/adafruit/Adafruit_TinyUSB_Arduino and install it manually. If you do not know how to do a manual install, see the link at at the bottom of this page.

If you're using the infrared control option you will also need to install IRLib2. It is not available directly through the Arduino IDE so you'll have to download it from GitHub and install it manually.

Installation of the IRLib2 library is as follows:

1. Visit the IRLib2 page on GitHub.
2. Select the “Download ZIP” button, or simply click this link to download directly.
3. Uncompress the ZIP file after it's finished downloading.
4. The resulting folder should be named "IRLib2-master" and will contain 5 separate folders. That is because IRLib 2.x is actually a collection of 5 libraries that work together. Sometimes in Windows you’ll get an intermediate-level folder and need to move things around.
5. Copy all five folders into your Arduino library folder alongside your other Arduino libraries, typically in your (home folder)/Documents/Arduino/Libraries folder. Libraries should not be installed alongside the Arduino application itself.
6. Re-start the Arduino IDE if it’s currently running

This repository consists of a total of five libraries each of which must be in your arduino/libraries/ folder. So for example it should be installed as follows...

- arduino/libraries/IRLib2
- arduino/libraries/IRLibFreq
- arduino/libraries/IRLibProtocols
- arduino/libraries/IRLibRecv
- arduino/libraries/IRLibRecvPCI
Do not install them in a single folder such as this...

- arduino/libraries/IRLib2_master
  - IRLib2
  - IRLibFreq
  - IRLibProtocols
  - IRLibRecv
  - IRLibRecvPCI

Here is a tutorial (link) that walks through the process of manually installing libraries that are not available through the library manager.

## Sample Code and Audio Files

The Arduino sketch to operate the clock along with the audio files are all available in a GitHub repository at [https://github.com/cyborg5/NeoPixel-Clock](https://github.com/cyborg5/NeoPixel-Clock). You can click on that link or click the green button below to directly download a ZIP file containing the files. This repository also contains the 3D model of this project in Fusion 360 format as well as STL files for 3D printing the case.

[Code and audio files on GitHub](https://github.com/cyborg5/NeoPixel-Clock)

Download the repository and unzip it. Although we will be able to drag and drop files onto the device as it was a USB drive, you have to load the initial set of MP3 files onto the SD card before we upload the software. Insert a formatted SD card into your PC and drag-and-drop all of the files from the "audio_files" onto the SD card. Do not drag the folder itself, just its contents. After the files are copied, safely eject it from your PC and insert it into the Music Maker Wing. It is safer to do this with the power disconnected. After the card is in place, connect a USB cable and plug it into your PC so that you can upload the code. Open the "clock.ino" sketch in your Arduino IDE. Do not upload it immediately because there is some configuration that you have to do first.

There are six tabs for the six files of this program. The first tab is "clock" containing the main program. Below is a partial listing of the beginning of that file. You should edit it to configure which options you have implemented. Use a "0" to disable the particular option or "1" to enable it.

```c
/*
 * Talking Musical Neo-Pixel Clock with Infrared, BLE, and Touch Controls
 * by Chris Young
```
Once you have configured the sketch for the options you have implemented, there is one additional step before you can upload the sketch. On the Arduino IDE under the "Tools-> USB Stack" menu you must select "TinyUSB" as seen in the image below. This enables the ability to drag-and-drop additional MP3 files onto the device as if it was a flash drive. After setting this option go ahead and compile and upload the sketch. If you get compiler errors, it is likely that you have not properly installed one of the required libraries.

When the system initializes you should hear a voice say "System Initialized... Right channel test... Left channel test". You should hear the first phrase from both speakers.
The other phrases in only one speaker. This will allow you to verify that both speakers are working. If you’re using the BLE option, after a few seconds you will hear "BLE initialized".

The clock initializes to 12 noon. The hours are depicted by a set of five blue or cyan colored pixels. Blue pixels indicate a.m. and cyan colored pixels indicate p.m. The minutes are depicted by a red pixel. The seconds are indicated by a moving green pixel. In the next section we will describe how to set the time and configure other options using the voice prompt setup menu.

If you have set USE_USB 1 then once the sketch is uploaded you should also see a new USB drive available on your PC. It will contain all of the MP3 files used by the system. If you ever need to update these files, you can simply drag-and-drop new files onto the drive and they will be used. You probably will not want to change anything except the music files. But if you want to record your own voice prompts or change the chimes you can do that as well. If you have no need of this feature you can disable it by changing the define to USE_USB 0 and re-upload the sketch.

NOTE: If you are using the touch controls, you may need to calibrate them. See the section on "Advanced Configuration" for details on how to do this.

Operating the Clock

There are 4 different ways that you can control the clock: infrared, BLE, touch controls, and serial monitor debugging. We will describe how each of these controls work and then explain how to use them. The methods of configuring the time and setting the options all work the same regardless of which kind of controls you use. Each control method has 7 options: up arrow, down arrow, right arrow, left arrow, set up, play, and reset.

Infrared Controls

For now we are assuming you are using the recommended Adafruit Mini Remote control. The software is already configured to use that kind of remote. If you are be using a different infrared remote, you will need to configure the software and recompile it to handle the particular IR protocol and code values for your remote. See the section on Advanced Configuration for details on how to do this. The image below shows the remote. The arrow buttons and "SETUP" buttons are obvious. The "PLAY" function is the "play/pause" red button at the top center. The reset button is the curved arrow button just above "3". Pressing any other button is ignored.
BLE Connect

You can use the Adafruit Bluefruit LE Connect app to connect your clock. You can download it from the iOS App Store or for Android from the Play Store. Complete details on this app can be found at this learning guide for the iOS version. The Android version works virtually the same.

When you open the app you will see a list of Bluetooth capable devices in your vicinity as shown below. Tap on the "Connect" next to "NeoPixel-Clock" as shown in the screen grab below.
When you have successfully connected, the clock should speak "BLE connected". You will then see the screen shown below. Tap on the "Controller" option third item from the bottom.
Then you should tap on the "Control Pad" option second from the bottom as shown below.
The control pad looks like the screen grab shown below. Obviously the arrow keys are used. Button 1 is "Setup". Button 2 is "Play". Button 3 is "Reset". The fourth button is not used.

If you disconnect your device from the clock you will hear "BLE disconnected" spoken.
Touch Controls

We have only implemented 4 buttons for the touch control. They are Setup, Right Arrow, Up Arrow, and Down Arrow. You may have to calibrate these controls as described in the Advanced Configuration section of this tutorial.
Serial Monitor Debugging Controls

If you recompile and upload the program using debugging mode you can open your serial monitor on your Arduino IDE and type commands into it to control the clock. Of course it must be connected to your computer and not operated off of an independent power supply.

Open up the program in the Arduino IDE and online and change it to read

```c
#define MY_DEBUG 1
```

Open your serial monitor and compile and upload the program. For every spoken feedback that the clock generates you also see a text version of those words appear on your monitor. You can type in a character into the serial monitor and press enter to control the clock. The commands are as follows: "U", "D", "R", and "L" for the arrow keys Up, Down, Right, and Left respectively. Use "S" for Setup and "P" for Play. Use "T" for Reset.

You can string commands together so for example you can type "RRRRR" and then press enter in the serial monitor to send the right arrow command five times.

Setting the Time

The arrow keys are used to set the time on the clock. The left and right arrows change the hours. The up and down arrows change the minutes. As you press these keys, the display on the clock will change and there will be audio feedback. There is no left arrow key when using touch controls so you may have to press the right arrow up to 23 times to set the hours. There is no Play or Reset functions available when using touch controls.

The Play button causes the clock to speak the current time. It will also play an animation on the NeoPixel display if you have animations enabled. It will also play music if you have music playback enabled.

The Reset button causes the seconds to reset to 59 seconds. You can use this to precisely synchronize the clock. We chose to have it reset to 59 seconds so that if you want to demonstrate the various features of the clock you could for example set it to 3:59 PM and then hit the reset. One second later it would click over to 4:00 PM and would perform chimes, animation, and music playback.
Here is a YouTube video that demonstrates how to set the time on the clock, demonstrates various features, and demonstrates the setup menu which is described on the next page.

The Setup Menu

When you press the Setup button the clock will go into a series of audio prompts that allow you to change various features.

It will begin by saying "Entering Setup Mode... press up or down arrow to change each item or press right arrow to continue to next item."

Here is a description of each of the items in the order in which they are presented.

Audio Mode

The clock has the ability to reduce the audio volume during nighttime. You have 2 different ways of specifying what is considered "nighttime". One of them is "time-based". In this mode you will specify the hour in which daytime begins and the hour in which nighttime begins.

The other method is to use the photocell. The photocell takes a reading and if it is below a certain level of ambient light it is considered "nighttime". See the Advanced Configuration section for details on how to calibrate the photocell. The threshold is set in the program and cannot be changed without editing and recompiling. It is not changeable via the options menus.

The Audio Mode menu has 4 options: disabled, continuous, time-based, or light-based. In a later menu item you will be given a chance to set the daytime volume level. You have selected "Continuous" then this value is used all the time. If you have selected either "time-based" or "light-based" then you also be given an opportunity later to set a nighttime volume level. If you have not implemented the photocell option, you will not be giving the light-based option in the menu system.

Brightness Mode

Similar to Audio Mode, the brightness mode allows you to have different brightness levels for daytime or nighttime using the same two options either time-based or light-based. There are three possible settings for Brightness Mode. They are constant, time-based, or light-based. The same definitions of daytime and nighttime whether
they are time-based or light-based apply to both the Audio Mode and Brightness Mode.

Daytime Begins and Nighttime Begins

If you have selected time-based for either the Audio Mode or Brightness Mode you will next be given the options to set the start hour for daytime and nighttime. Pressing up or down arrows changes the hour. If you have not selected any time-based options then this section is skipped.

Daytime Volume Level

The next option is the daytime volume level. This is the default level if you have not defined a time-based or light-based Audio Mode so you will always be given this option. As you press the up or down arrows it gives you a numeric feedback of the value used. Note that the Music Maker Wing uses small numbers for high-volume and large numbers for low volume. You will be able to hear the volume level as you adjust it. Because it would take a long time to reach 255 which is the lowest volume level available, we have defined anything past 50 as being muted. The default value is 20 and we recommend not setting it no louder than 10 although it can go all the way to a maximum value of 1.

Nighttime Volume Level

If you have specified Audio Mode either time-based or light-based you will be given the option to set a nighttime volume level. Again up and down arrows change the value and you will be able to hear the volume level you have set. Lower numbers are louder and higher numbers are softer with limits of 50 through 1. If you have not selected one of these audio modes, this option is skipped.

Daytime Brightness Level

This allows you to adjust the brightness level of the NeoPixel display. This is the default if you have selected "continuous" Brightness Mode or it is the daytime level if you have selected time-based or light-based. You will hear a numeric value as well as visibly see the brightness change. Keep in mind that if you use extremely high brightness it will draw more current and may exceed the capacity of your power supply.
Nighttime Brightness Level

If you have selected a Brightness Mode of time-based or light-based then you will be given the option to select a nighttime brightness level. You are not only hear the numeric value but will be able to see the brightness level you are setting. If you have selected continuous Brightness Mode then this option is skipped.

Chimes Option

There are three settings for the chimes. They are disabled, Westminster, or cuckoo clock. Westminster chimes are patterned after those used by Big Ben in London England. At the quarter hour there are four notes. At the half-hour there are eight notes. At the three-quarter hour there are 12 notes and at the top of the hour the full 16 note melody is played as well as a gong counting the hour from 1 to 12. For more information on Westminster chimes see this article on Wikipedia. The other option is a cuckoo clock sound effect. There are no quarterly chimes for the cuckoo clock option. At the top of the hour you will hear a gong and a "cuckoo" sound effect for each hour from 1 to 12.

Voice Announcement

You can have the clock speak the current time at each of the four quarters of the hour. This feature can be enabled or disabled. Note that if it is disabled you can still get the clock to speak the current time whatever time it is by pressing the "Play" button.

Hourly Audio

You can enable or disable either the chimes and voice announcements that occur at the top of the hour by changing this option. Note this does not affect music playback. That is configured in a different option.

Quarterly Audio

You can enable or disable the chimes and voice announcements that occur at the four quarters of the hour. When using the Westminster chimes if you have Hourly Audio Enabled but Quarterly Audio Disabled then you will not hear the 16 note melody but you will get the hourly gongs from 1 to 12.
Animation Mode

You can have the NeoPixel display a pattern colors at the top of the hour. We have programmed in 12 different patterns. These three modes available are: disabled, sequential, and random. The sequential mode always displays the same pattern based on the hour. Random mode displays a random pattern. If you press the play button and this is either sequential or random you will also see the animation based on the hour at which press button.

Music Mode

You can have the clock play a musical clip at the top of the hour or when you press the play button. They are the files "music01.mp3" through "music12.mp3". We have selected brief clips from the Adafruit music album "Frequency" by Bartlebeats. This is the background music that Adafruit uses to introduce its YouTube videos and live streams. You can find the full album here [https://bartlebeats.bandcamp.com/album/frequency](https://bartlebeats.bandcamp.com/album/frequency) or even order a vinyl version here. You can substitute your favorite MP3 files by renaming them and copying them to the SD card. For example you might want to put holiday music on at that time of the year. There are three modes for this: disabled, sequential, and random. Sequential plays the corresponding music clip based on current hour. Random plays a random clip. This also changes which musical clip is played when you press the play button.

Alarm Mode

You can set an alarm at any particular hour and minutes. The modes are enabled and disabled. If enabled, you will then be given the opportunity to set the alarm hour to any of 24 hours and minutes. When that particular hour is reached, the clock will play "alarm.mp3"

This completes the setup menu and you will hear a voice say "Setup completed". Here is what the setup process looks like with debugging turned on and the serial monitor open. We skipped through all of the defaults until we got to the music mode and then returned it off.

Debug output ready.
System Initialized
Entering setup mode
Press arrows to change or advance to next item
Audio mode is currently time based.
Brightness mode is constant.
Daytime begins at: 10 a.m.
Nighttime begins at: 10 p.m.
Daytime volume: 20
Nighttime volume: 40
Daytime brightness: 6
Chimes are Westminster chimes.
Voice announcement enabled.
Hourly audio is enabled.
Quarterly audio is enabled.
Animation mode is currently random.
Music mode is currently random.
Music mode is currently disabled.
Alarm is enabled.
Alarm hour is 7 a.m.
Alarm minutes is 30
Setup completed

3D Printed Case

We have designed a 3D printable faceplate and circular frame that features an optional hanger for hanging the clock on the wall or a stand for sitting it on a table. Of course you are free to use these electronics and programs and files with whatever kind of case you might want to design for yourself. You could use wood, metal, cardboard or whatever materials you want.

The frame is just over 239 mm in diameter so it will not fit on many normal size 3D printers. We were able to print ours in a single piece on a PrintrBot Metal Plus with a 254 x 254 x 254 build area. We have provided alternate parts that are split into more manageable size but we have not test printed these parts so you may need to play with the tolerances a bit. We have provided Fusion 360 files that you can modify yourself as well as a complete set of STL files that you can print directly.

The faceplate has 60 holes for the NeoPixels to shine through as well as 2 holes for the IR receiver and the photocell. We found it a bit difficult to print the first layer with all of these tiny holes so we provide the STL files with the holes filled in on the first layer. You can then take a drill or a Dremel Tool to drill them out. If you have especially good bed adhesion you might be able to print with the holes already cleaned out. We will show you how to modify the Fusion 360 model to clean out the holes. We will also show you how to modify the Fusion 360 model to either split the pieces or print them as one unit.

Although we have shown the renderings with a white faceplate so that is easier to see the details, we actually printed ours in black PLA. We printed the circular frame in white PLA and then spray-painted it with a copper color spray-paint.
All of the files for 3D printing are in the "3D_models" folder that you downloaded with the software for this project. We will also make them available on Thingiverse at this link [https://www.thingiverse.com/thing:3025771](https://www.thingiverse.com/thing:3025771).

The faceplate comes in five pieces. A center, top, left, right, and bottom. The joint between each piece has a dovetail notch. Print all five pieces and then assemble them on a flat surface. We recommend gluing them together with superglue or other adhesives. Similarly the frame ring comes in 4 pieces and are tied together with dovetail notches. When printing the ring you will have to flip the orientation of the STL files 180° so that the flat surface is on your print bed. The top and bottom pieces of the ring have notches cut in them so that you can later slide in the wall hanger or the table stand. You may need to use support material to print these notches successfully. We also provide STL files for the complete faceplate as a single piece and the complete ring as a single piece if your 3D printer has a large enough build surface. The hanger can print in its original orientation. You will have to rotate the stand 90° so that it's under surface is flat on the print bed.

The image below shows the four printed parts as we printed them with the faceplate and ring as solid pieces. We printed with 2 bottom layers, 3 top layers, 2 shells with 30% infill using PLA.
Assembling the Clock

Because it might be difficult to get the IR receiver and the photocell to lineup in the holes properly, you may want to print just the center section of the faceplate and test fit the assembled clockworks on that piece. Drill out the holes for the IR receiver and the photocell. Then lineup the clock works and see if you can get a clear view of the IR receiver photocell through the holes. If you need to adjust them, you may be able to edit the STL files using Blender 3D or some other STL file editor. Or you can go back to the original Fusion 360 model to modify it and export new STL files. See the section on "Modifying the Model" for details on how to do this. Even if you are printing the faceplate in one piece, it might be a good idea to print just the center section as a test to make sure the holes align properly.

The holes for the LEDs should already be spaced properly and will not need any modification. As mentioned previously, we left the holes partially filled in to facilitate printing but if you want to modify the model to have the holes print without drilling we will show you how in the section on "Modifying the Model".

Assemble the five pieces of the faceplate by snapping them together on a flat surface and gluing them together with superglue or an adhesive of your choice. Drill out the 60 holes.

Mount the Speakers

We recommend that you mount the speakers first using two screws and nuts for each speaker using the mounting tabs we have provided. You should temporarily disconnect the speaker wires from the Music Maker Wing while mounting the speakers. We recommend you mount them first before mounting the clockworks so you can have easier access to put in screws and nuts. If you don't have the proper size screws you could glue them in place. Orient them so that the speaker wires are coming out of the bottom of the clock. That is towards the end where the touch plate area will be.
Mount the Doubler Board

Remove the Feather M0 and Music Maker Wing from the doubler board so that you can get access to the mounting holes. Align the board over the 4 standoffs in the center of the plate. The NeoPixel wires should extend towards the top of the clock and the four touch control wires should extend towards the bottom. We recommend using M3 plastic screws just to ensure that it doesn’t accidentally shorten something out. If

Touch Controls

Insert 4 metal screws into the touch control area of the plate. Strip the ends of the wires and either wrap them around the screws and solder them or solder them onto the tops of the screw heads. They must be metal for the capacitive touch feature to work.

NeoPixel Ring

Plug-in the Feather M0 and the Music Maker Wing into the doubler board. Mount the NeoPixel ring onto the plate. There is a sample sketch in the files you downloaded called "lamp_test". Plug a USB into the Feather M0 and upload that sketch. It will light
all of the NeoPixels as white at a reasonable brightness. The topics of will be colored red. This will help you align the ring so that the pixels are visible through the holes in the plate.

If you don't get the "zero" pixel at the very top, don't worry there is an option in the software that allows you to specify which pixel is the top zero minute pixel. See "Advanced Configuration" for details.

We put a dab of hot glue at all four positioning tabs around the ring. We used hot glue rather than superglue in case we got the ring in the wrong position it would be easier to break it loose and reposition it.

As a final step, connect the speaker wires back onto the Music Maker Wing terminals with the red wires on the outside and the black wires towards the center. You may wish to shorten these wires as well as shortening the wires connecting to the NeoPixel ring.

Frame Ring Assembly

Assemble the 4 pieces of the frame ring. We suggest superglue or other adhesive. You can slide the wall hanger into the notch at the top of the ring and optionally add the table stand into the notch at the bottom of the ring. Then insert the plate into the front of the frame and glue it in place. Your project is now complete. Upload the clock sketch and try out all the options.
Below is a photograph of our actual completed project. The faceplate could probably benefit from some filling, priming, sanding and painting. However from a distance under normal lighting conditions you don't see any of the artifacts that are apparent in this photo.
Modifying the Models

About Fusion 360

This entire project, not just 3D printed portions, was modeled using Autodesk Fusion 360. The renderings and screen grabs you have seen throughout this tutorial were all created using the software. Although it can be quite expensive to purchase a license, it is available for free use for students and for hobbyists and companies making less than $100,000 per year. It is a powerful CAD program that is especially well-suited to 3D printing. The most powerful part of the program is that it remembers each step used to create the design and allows you to go back in time to edit earlier steps. The effect of those edits propagate forward into the later steps. It also allows you to parameterize your design with user-defined numeric values that can be easily edited. When you change these values, everything dependent upon those values changes as well.

Loading the Model

In the files you downloaded for this project in the "3D_models" folder there is a file titled "NeoPixel Clock.f3z". Open up Fusion 360 and create a new project. Inside the
empty project at the top of the screen you will see an "Upload" button. We will need to upload this design to your account. All files in Fusion 360 are stored in the cloud so we will have to upload them to the cloud to use them. When the files have been uploaded you will see several objects. This model consists of a main model called "Clock" which contains links to other components. You only need to open the main model to make changes.

1. Create a new project from your main account menu.
2. Upload the model to the cloud.
3. Open the "Clock" model.
User Defined Parameters

We have added several user-defined parameters that allow you to easily tweak tolerances and features of the model. You access the User-Defined Parameters under the Modify toolbar. It is the last option "Change Parameters". You will see the following pop-up appear.

Of particular interest would be the values ScrewHole which is the diameter of the mounting holes for the doublet board. SensorHoles is the diameter of the holes for the IR sensor and photocell. The values IRx and IRy are the position of the hole for the IR sensor. Similarly PCx and PCy are the position of the Photocell hole. SplitTolerance is the gap that is cut to split the faceplate and frame ring into multiple pieces. If you have trouble assembling the pieces you might want to increase this value. You can click on any of these parameters, change the value, and the model will update automatically. Then you can export the pieces as STL files and print them again.

Split versus Solid

As mentioned before, one of the advantages of Fusion 360 is that it keeps a timeline at the bottom of the screen that keeps track of the steps necessary to create your design. By editing this timeline, you can make changes to the model and everything will update automatically assuming you don’t make too many drastic changes that ruin future references.
As we have constructed this model, the final two steps of the design are to split the ring into four pieces and to split the faceplate into five pieces. If you backup the timeline 2 spots, it will undo these changes and you can export these pieces as single solid pieces presuming your 3D printer is capable of printing such large pieces.

**Filling the Holes**

We had difficulty printing the faceplate with the tiny holes. The 3D printer draws the little rings around the holes and then tries to fill in the spaces around them. But if those tiny little rings don't adhere perfectly, everything can get messed up. We chose to create a one layer plug 0.1 mm thick that will fill in the NeoPixel holes and the sensor holes for the IR sensor and photocell. You then later have to drill out these holes. If you want to try to print the parts with the holes predrilled you can modify the model to eliminate these plugs and export the STL files again.

On the left side of the screen where all of the components are listed, hover your mouse over the "plate" and then move slightly to the right and click on small circle that will appear. This will limit your timeline to only the steps necessary to construct the plate. Then look at the bottom of your screen in the timeline. If you hover your mouse over each of the steps you can see what it does. The second from the final step in creating the faceplate is to fill in the 2 sensor holes. The third from the final step is to fill in the 60 NeoPixel holes. You can right-click on these steps and delete or modify them to suit your needs.

When you have finished making these modifications, go back to the top of the list of components and select the entire model by hovering your mouse over the top item and then clicking on the small circle to the right of it. Save your file after making these modifications.
In order to export the modified file as an STL file, click on the "Make" button in the upper right corner of the toolbar. You will see a pop-up on the right. You should uncheck "Send to 3D print utility". Then click on the body that you wish to export. For most of the model you can use medium output but for the ring to get the best quality of the curve we suggest using high output in the drop down box. When you click okay you will be given a dialogue to save the file to your hard drive.
Advanced Configuration

We have provided lots of ways to reconfigure the software depending on the options you've chosen when building this project and we have provided lots of ways to set various defaults. You may also need to calibrate the photocell to set the threshold for what is considered "day" or "night". You may also need to calibrate the touch controls because depending on the size or type of screw you have used, the length of the wires, the method used to connect them it can vary the capacitance of that system. If you want to use the infrared controls using your own remote control rather than the recommended Adafruit Mini Remote you will need to configure that as well.

We have already told you how to turn off and on various features of this project such as the touch controls, IR sensor, photocell etc. in this section "Sample Code and Audio Files". Here are the remaining configuration changes you can make.

Defining the Top Pixel

If you happen to mount your NeoPixel ring or the faceplate at a slight angle, you may not end up with the "0" pixel exactly at the top. On line 73 of the main sketch tab you can modify the following line to change the top pixel.
Calibrating Touch Controls

The touch controls work by measuring the electrical capacitance of the wires connected to pins A0 through A3. It uses the Adafruit_FreeTouch library. Normally there will be a low capacitance read from those pins and when you touch them with your finger you will get a higher reading. You have to set a threshold which defines when the pin has been touched. These values can vary even from pin to pin in your project. It depends on the length of the wires, how they are routed past other components, the types of screws to which you have attached the wires, and how you attached them. To calibrate this system, there is a sample sketch called "freetouch_test" included in your package of programs. You should open your serial monitor and upload this sketch. It will print out 4 values showing the current readings five times per second. You might also want to try opening the plotter function of the Arduino IDE to get a graphical representation of the data as you touch each of the pins.

As you can see from the graph, there is a wide variety for the upper and lower values of each of the four buttons. We will have to pick different threshold values for each one. Once you have an idea of a good threshold that accurately describes when you have touched the pins, you can modify the values in the original "clock" sketch at approximately line 94.

```cpp
#if(USE_TOUCH)
  #include "Adafruit_FreeTouch.h"
  #define TOUCH_SETUP_THRESHOLD 650
  #define TOUCH_RIGHT_THRESHOLD 550
```

©Adafruit Industries
#define TOUCH_UP_THRESHOLD 550
#define TOUCH_DOWN_THRESHOLD 750

Additional configuration changes you may need to make begin on line 32. We have defined which pin numbers correspond to the various touch controls. If you have wired up your controls in a different order than we have, you might need to change these pin numbers.

#define TOUCH_SETUP_PIN A3
#define TOUCH_RIGHT_PIN A2
#define TOUCH_UP_PIN A1
#define TOUCH_DOWN_PIN A0

**Initial Defaults**

Beginning at line 38 there are series of defines that are the values used in the menu system and other startup values. The first few lines define the various modes used for animation, audio settings, chimes etc. The following lines can be edited to set up the default values when the clock is first powered up. Once you've played around with the various features of the clock, you will probably want to change these defaults to the values you prefer and then recompile and upload so you don't have to change all of these values each time the clock is powered up.

```
//Audio, Music and Animation modes
#define MODE_OFF 0
#define MODE_ON 1
#define MODE_TIMED 2
#define MODE_LIGHT 3
#define MODE_RANDOM 4
#define MODE_WESTMIN 5
#define MODE_CUCKOO 6

//Initial default settings. These values are set in "audio_menu.h" and other places // but the defaults are defined here for easy access.
#define AUDIO_DEFAULT MODE_TIMED
#define ANIMATION_DEFAULT MODE_RANDOM
#define MUSIC_DEFAULT MODE_RANDOM
#define CHIMES_DEFAULT MODE_WESTMIN
#define BRIGHT_MODE_DEFAULT MODE_ON
#define VOICE_DEFAULT true
#define QUARTERLY_DEFAULT true
#define HOURS_DEFAULT 12
#define HOURLY_DEFAULT true
#define MARKS_DEFAULT true
#define ALARM_DEFAULT true
#define ALARM_HOUR 7
#define ALARM_MINUTES 30
#define DAYTIME_START_DEFAULT 10
#define NIGHT_begINS_DEFAULT 22
#define VOLUME_DAY_DEFAULT 20
#define VOLUME_NIGHT_DEFAULT 40
#define VOLUME_THRESHOLD 50
#define BRIGHT_DAY_DEFAULT 6
#define BRIGHT_NIGHT_DEFAULT 2
#define HOURS_DEFAULT 12
```
Changing IR Codes

The infrared receiver uses IRLib2 to receive and decode infrared signals. This powerful open source library supports 11 different popular IR protocols and a variety of variations of those 11 protocols. We've already configured it for the Adafruit Mini Remote which uses the popular NEC protocol. However you can use any remote want if it uses a protocol that is supported by IRLib. Obviously you don't want to pick a remote protocol that is going to interfere with any consumer electronic devices you have. Otherwise every time you turn up the volume on your TV, the clock with change. Pick a device that will interfere. For example my clock is in my office but my Blu-ray player is in my living room. I could use the Blu-ray codes to set the clock. If you have universal remote, perhaps program one of the device sections for some device do not own. Any of the 11 protocols supported by IRLib2 should work however we recommend against using Sony protocol because it automatically sends three copies of every signal when you push the button. That might give you double or triple signals where you don't want them.

There is an extensive user's manual for IRLib2 available in the "libraries/IRLib2/manuals" folder of the library. It is available in Microsoft Word, PDF, and EPUB formats.

In the "clock" program at line 82 you will see the following section which sets up IRLib2 to use the NEC protocol and to define the codes for the Adafruit Mini Remote.

```c
#if(USE_IR)
    //set up IRLib2 to use NEC protocol and Adafruit mini-remote
    #include "IRLibRecvPCI.h"
    #include "IRLibDecodeBase.h"
    #include "IRLib_P01_NEC.h"
    #include "adafruit_mini_codes.h"
    IRrecvPCI myReceiver(IR_PIN);
    IRdecodeNEC myDecoder;
#endif
```

Temporarily comment out the three #include statements and add a new statement to include "IRLibAll.h". Also change the class type of the decoder from "IRdecodeNEC" to be just "IRdecode". This will enable IRLib2 to decode not just NEC protocol but all of the protocols it supports. The code should now look like this...

```c
#if(USE_IR)
    //set up IRLib2 to use NEC protocol and Adafruit mini-remote
    //#include "IRLibRecvPCI.h"
    //#include "IRLibDecodeBase.h"
    //#include "IRLib_P01_NEC.h"
    #include "adafruit_mini_codes.h"
    #include "IRLibAll.h"
```
IRrecvPCI myReceiver(IR_PIN);
IRdecode myDecoder;
#endif

At approximately line 288 you will see a debugging section that includes a statement to dump details of the code received by IRLib2. You should uncomment this statement and also turn on debugging at the top of the sketch.

/*
 * Talking Musical Neo-Pixel Clock with Infrared, BLE, and Touch Controls
 * by Chris Young
 * released to the public domain
 */
//Various options. Set to zero to disable or one to enable.
//Enables Serial Monitor Debugging
#define MY_DEBUG 1

//... Skip down to line 288
#if (MY_DEBUG)
  myDecoder.dumpResults(false);
#endif

Compile and upload the sketch with your serial monitor open. Try pressing various buttons on your remote control such as the arrow keys, set up button, play button, and some sort of reset or exit button to be used as the reset function of the clock. We tried using our Blu-ray remote control. We pressed the up, down, left, right, menu, play, and exit buttons in that order. We get the following output on the serial monitor.

Debug output ready.
System Initialized
Decoded Samsung36(8): Value:E18E7 Adrs:400 (36 bits)
Decoded Samsung36(8): Value:E9867 Adrs:400 (36 bits)
Decoded Samsung36(8): Value:ED827 Adrs:400 (36 bits)
Decoded Samsung36(8): Value:E58A7 Adrs:400 (36 bits)
Decoded Samsung36(8): Value:E6897 Adrs:400 (36 bits)
Decoded Samsung36(8): Value:E28D7 Adrs:400 (36 bits)
Decoded Samsung36(8): Value:ED42B Adrs:400 (36 bits)

This tells that it successfully decoded Samsung36 protocol which is protocol number 8 in the IRLib2 system. It is a 36 bit protocol consisting of two parts. An Address of 0x400 which seems to be constant and a Value of a five digit hexadecimal number that changes depending upon the function. We can now plug in these values into the sketch beginning at approximately line 274 change the code to read as follows:

#if(USE_IR)
  if (myReceiver.getResults()) {//was a command received by IR?
    if (myDecoder.decode()){//is that the right protocol?
      switch(myDecoder.value) {
        case 0xE18E7: Value= UP_ARROW; break;
        case 0xE9867: Value= DOWN_ARROW; break;
        case 0xED827: Value= LEFT_ARROW; break;
        case 0xE58A7: Value= RIGHT_ARROW; break;
        case 0xE6897: Value= SETUP; break;
        case 0xE28D7: Value= PLAY; break;
        case 0xED42B: Value= RESET; break;

©Adafruit Industries
Once you have verified that everything is working, you can comment out the "myDecoder.dumpResults(false);" line above and turn off debugging at the top of the sketch. Currently we have IRLib2 configured to receive all protocols which is quite wasteful, now that we know that we use protocol 8 which is "Samsung36" we can modify IRLib include files to use just that protocol. Change line 83 and beyond to read as follows...

```c
#if(USE_IR)
  //set up IRLib2 to use NEC protocol and Adafruit mini-remote
#include "IRLibRecvPCI.h"
#include "IRLibDecodeBase.h"
#include "IRLib_P08_Samsung36.h"
IRrecvPCI myReceiver(IR_PIN);
IRdecodeSamsung36 myDecoder;
#endif
```

This code looks very much like the original version except that we have substituted "IRLib_P08_Samsung36.h" in place of the previous include file for the NEC protocol. We have also modified the decoder so that it uses the "IRdecodeSamsung36" class. We have also removed the reference to "adafruit_mini_codes.h" which we are no longer using.

If your remote is not using one of the protocols supported by IRLib2 there is one other alternative. There is a hash code decoder available that will attempt to generate a unique hash code based on your button presses. It will attempt to generate a unique 32-bit hex value based on your button press. This is not as accurate or efficient as using the actual protocol but it will work in a pinch. To use the hash decoder, change the include section at line 83 to read as follows:

```c
#if(USE_IR)
  //set up IRLib2 to use NEC protocol and Adafruit mini-remote
#include "IRLibRecvPCI.h"
#include "IRLibDecodeBase.h"
#include "IRLib_HashRaw.h"
#include "adafruit_mini_codes.h"
IRrecvPCI myReceiver(IR_PIN);
IRdecodeHash myDecoder;
#endif
```

Then try pushing buttons on a remote, recording the 32-bit hex values and substituting them into appropriate portion of the sketch as we did before.
Calibrating the Photocell

If you’re using a light-based option to define day and night for the audio volume or NeoPixel brightness level you may need to change the threshold value that determines what given amount of ambient in the room constitutes "night".

There is a sample sketch included with the package titled "photo_test" which will allow you to verify that the photocell is working. Upload the sketch and open the serial monitor. Cover or uncover the photocell and watch the values change. Once you know it is working you can use the "clock" sketch with debugging turned on to collect data on ambient light values in your room overnight.

You should turn on debugging at the top of the sketch and open the serial monitor and upload. The sketch will print out the amount of light received by the photocell every 15 minutes. Leave it running overnight and then look at the results next day. This should help you determine an appropriate threshold level for your environment.

Here are the results from an overnight run in my home office.

```
Photocell value:535 The current time is 9:15 p.m.
Photocell value:536 The current time is 9:30 p.m. &lt;Office lights off
Photocell value:60  The current time is 9:45 p.m.
Photocell value:62  The current time is 10 p.m.
Photocell value:61  The current time is 10:15 p.m.
Photocell value:64  The current time is 10:45 p.m.
Photocell value:61  The current time is 11 p.m.
Photocell value:64  The current time is 11:15 p.m.
Photocell value:63  The current time is 11:30 p.m.
Photocell value:61  The current time is 11:45 p.m.
Photocell value:62  The current time is 12 midnight &lt;House lights off
Photocell value:9   The current time is 12:15 a.m.
Photocell value:10  The current time is 12:30 a.m.
Photocell value:9   The current time is 12:45 a.m.
Photocell value:9   The current time is 1 a.m.
--- Skipping ahead ---
Photocell value:9   The current time is 2:15 a.m.
Photocell value:23  The current time is 2:30 a.m. &lt;Late-night bathroom visit
Photocell value:0   The current time is 2 a.m.
--- Skipping ahead ---
Photocell value:9   The current time is 6:30 a.m.
Photocell value:12  The current time is 6:45 a.m.
Photocell value:11  The current time is 7 a.m.
Photocell value:27  The current time is 7:15 a.m.
Photocell value:46  The current time is 7:30 a.m.&lt;Daylight begins
ALARM!
Photocell value:31  The current time is 7:45 a.m.
Photocell value:47  The current time is 8 a.m.
Photocell value:43  The current time is 8:15 a.m.
Photocell value:47  The current time is 8:30 a.m.
Photocell value:69  The current time is 8:45 a.m.
Photocell value:68  The current time is 9 a.m.
Photocell value:60  The current time is 9:15 a.m.
Photocell value:75  The current time is 9:30 a.m.
Photocell value:77  The current time is 9:45 a.m.
Photocell value:76  The current time is 10 a.m.
```
<table>
<thead>
<tr>
<th>Photocell value</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>10:15 a.m.</td>
</tr>
<tr>
<td>82</td>
<td>10:30 a.m.</td>
</tr>
<tr>
<td>114</td>
<td>10:45 a.m.</td>
</tr>
<tr>
<td>543</td>
<td>11 a.m.</td>
</tr>
<tr>
<td>541</td>
<td>11:15 a.m.</td>
</tr>
<tr>
<td>552</td>
<td>11:30 a.m.</td>
</tr>
</tbody>
</table>

Based on this information we set the light threshold at 100 which can be found outline 70 of the clock sketch.

```
#define LIGHT_THRESHOLD  100
```