



SmartMatrix Remote Controlled LED Art Display

Created by Louis Beaudoin



<https://learn.adafruit.com/smartmatrix-remote-controlled-led-art-display>

Last updated on 2023-08-29 02:42:03 PM EDT

Table of Contents

Overview	3
<ul style="list-style-type: none">• Major Components• Parts List for Guide	
Assembly	6
<ul style="list-style-type: none">• SmartMatrix Shield• IR Receiver Cable	
Programming Aurora Sketch	12
<ul style="list-style-type: none">• Lite Tool Installation• Full Installation	
Using Arduino Sketch	13
<ul style="list-style-type: none">• Remote Control Buttons• Customizing the Display	
Troubleshooting	14
<ul style="list-style-type: none">• Nothing is displayed• "No SD card" message• GIFs aren't playing• Sketch does not respond to IR remote• "No Clock" message	
Acknowledgements	15

Overview

This guide will show you how to add an IR receiver to a Teensy 3.1 to play patterns on a 32x32 Pixel RGB LED Matrix Panel. The Teensy 3.1 runs an Arduino sketch that draws to the display, listens for IR remote commands, and refreshes the display at a high frame rate with good color depth. The SmartMatrix Shield makes it easy to connect everything together.

* This guide can be used in combination with the SmartMatrix Animated GIF Player guide. Adding a microSD card adds support for playing animated GIF files in addition to patterns in the Arduino Sketch.

We will explain the basic soldering and wiring to connect the IR receiver to the SmartMatrix Shield, and how to load the Arduino sketch that controls everything onto the Teensy 3.1.

The demo video shows the display mounted in a shadow box display with frosted acrylic diffuser (1/4" gap between LEDs and diffuser on the right, 1/2" gap on the left). The LEDs look better when diffused, whether its through a piece of copy paper or through specialty lighting acrylic. The frame is a subject for a future tutorial.

Major Components

32x32 RGB LED Matrix Panel

A 32x32 RGB LED Matrix Panel is an affordable way to add over a thousand bright pixels to your project. Compared to a Neopixel matrix, these panels use smaller LEDs which are bright but not as blinding, and have a higher pixel density. Unlike a Neopixel matrix, the panel requires a microcontroller to refresh the LEDs continuously to display an image. It takes about 40% of the CPU time and most of the memory of an Arduino Uno just to refresh a panel, and that's with a low refresh rate and a limited 12-bit color palette. To get a better image quality, we will use a more powerful microcontroller.

Teensy 3.1

Despite its small size, the Teensy 3.1 packs a lot of horsepower as it uses an ARM Cortex M4 CPU that runs at 72MHz and has 64kB of RAM. This is more than enough power and memory to refresh the display with a high refresh rate and full color palette. Even better, the processor in the Teensy 3.1 comes with a DMA engine that can move data from memory to pins to allow refreshing the display in the background, so the main program can run without as many interruptions. The Teensy 3.1 is

programmable using a modified Arduino IDE, and uses many of the same libraries as normal Arduino boards, so it should be familiar to program for anyone that has written an Arduino sketch before.

SmartMatrix Shield and Library

The SmartMatrix Shield is the best way to connect the Teensy 3.1 to the RGB LED Matrix Panel. The shield takes care of routing the 13 signals needed to refresh the display, adds convenient connectors for power, and brings the unused signals out to a convenient expansion header. The SmartMatrix Library includes code that refreshes the display in the background, and provides functions that make it easy to draw to the display and scroll text on top of the drawing.

Aurora Arduino Sketch

[Aurora \(\)](#) is an Arduino Sketch designed to give a lot of options for displaying customizable content on a 32x32 RGB Matrix Panel, with an easy to use interface controllable through an IR remote control. Using the remote you can choose a pattern or animation to display, change the brightness to match ambient light, sleep the display when you are not using it, and enable autoplay to rotate through content automatically.

IR Sensor, Remote, Arduino Library

To remotely control the Arduino sketch we will add an IR Sensor to the SmartMatrix Shield, and use a compatible remote control. The [Arduino-IRremote \(\)](#) library decodes the signal coming from the IR sensor.

FastLED Library

From the [project's homepage \(\)](#): "FastLED is a fast, efficient, easy-to-use Arduino library for programming addressable LED strips and pixels such as WS2810, WS2811, LPD8806, Neopixel and more. FastLED is used by thousands of developers, in countless art and hobby projects, and in numerous commercial products."

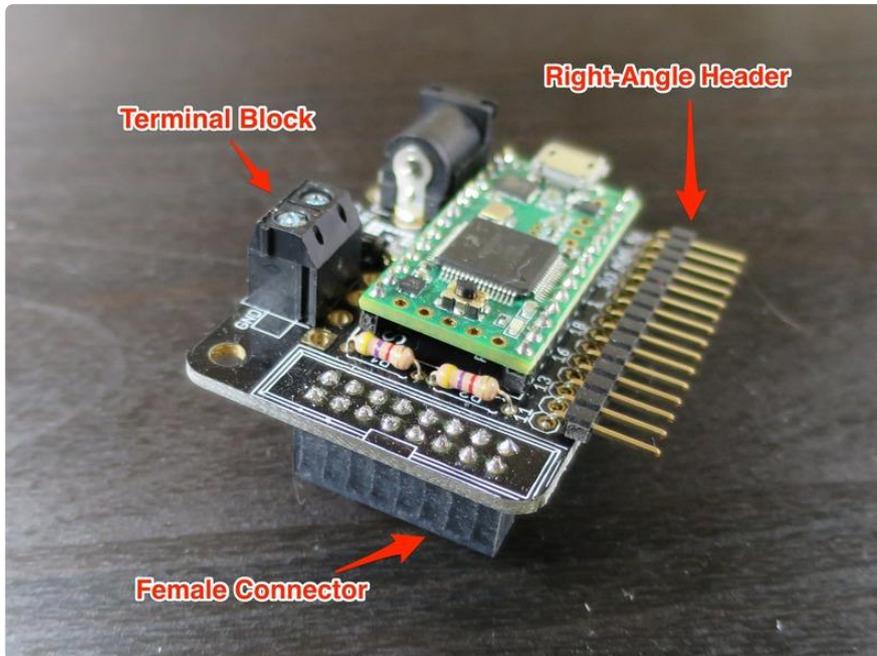
We use FastLED's helper functions to efficiently generate patterns and work with color palettes. FastLED has a great community, and many of the patterns in our Arduino sketch have been contributed by the [FastLED community \(\)](#).

Parts List for Guide

- [32x32 RGB LED Matrix Panel - 6mm pitch \(\)](#)
- [Teensy 3.1 \(\)](#)
- [SmartMatrix Shield for Teensy 3.1 \(\)](#)
- [5V 4A Switching Power Supply \(\)](#)
- [15-pin right-angle male header \(\)](#)
- [IR Remote \(battery included\) \(http://adafru.it/389\)](http://adafru.it/389)
- [IR Sensor \(http://adafru.it/157\)](http://adafru.it/157)
- 3-conductor Female/Female Jumper Wires: [3" \(\)](#) / [6" \(http://adafru.it/266\)](http://adafru.it/266) / [12" \(http://adafru.it/793\)](http://adafru.it/793)
 - Select long enough wires to mount the IR Receiver where you want it. 3" is just long enough to reach the closest edge of the panel.
- Needle Nose Pliers
- Optional for Animated GIFs
 - [MicroSD Memory Card and SD to MicroSD Adapter \(\)](#)
 - [6-conductor 3" Female/Female Jumper Wires \(\)](#)
 - MicroSD Reader to load files onto card
- For [SmartMatrix Shield Assembly \(\)](#):
 - Soldering Iron
 - Solder
 - Diagonal Cutters
 - Wire Strippers
 - Small Flathead Screwdriver
 - X-Acto or similar Knife
 - Recommended: Multimeter

Assembly

SmartMatrix Shield



We'll start with an assembled SmartMatrix Shield. For the pictures in this guide we used a SmartMatrix Shield with female connector to plug the Shield directly into the panel, the terminal block power option, and we added the diode and cut the Teensy VUSB trace to power the Teensy from the external power supply. You can use the other assembly options listed in the SmartMatrix Shield assembly instructions if you prefer.

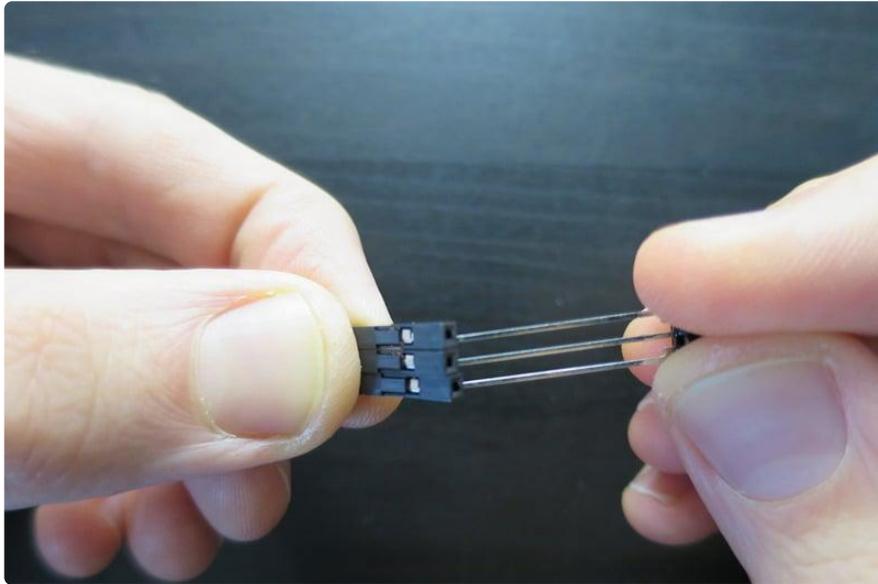
To make for a lower profile install, you might want to swap out the straight male pins for the expansion header with right angle pins if you have them, like we did in this guide.

Refer to [the instructions here \(\)](#) to assemble the shield.

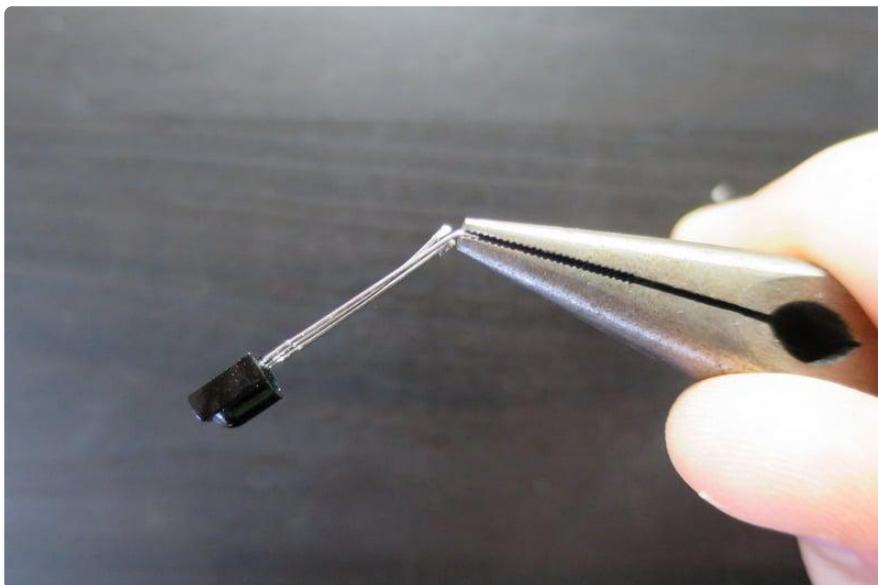
If you want to install the optional microSD card for playing animated GIFs, refer to the instructions in the [SmartMatrix Animated GIF Player guide \(\)](#). Follow all the instructions in the tutorial and make sure you are able to play GIFs. The Arduino sketch in this tutorial will read GIFs from the same directory on the microSD card.

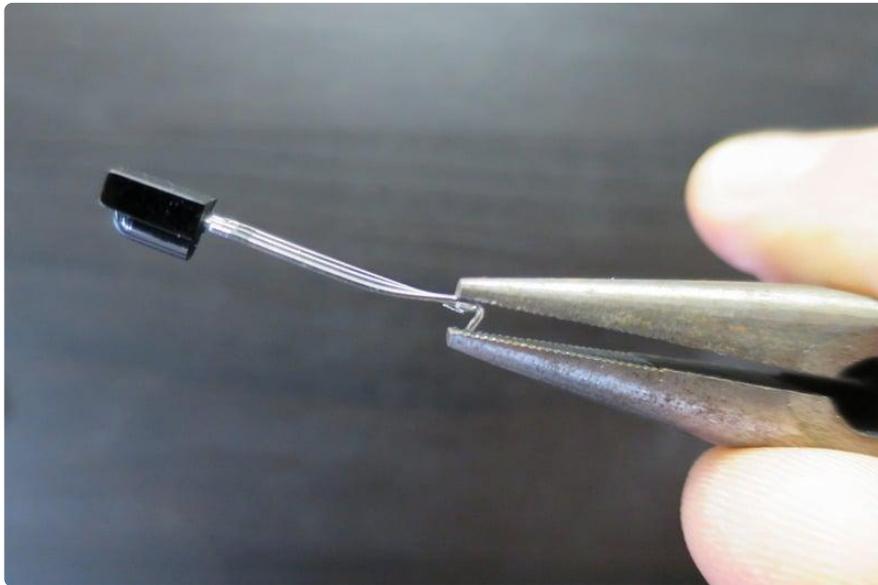
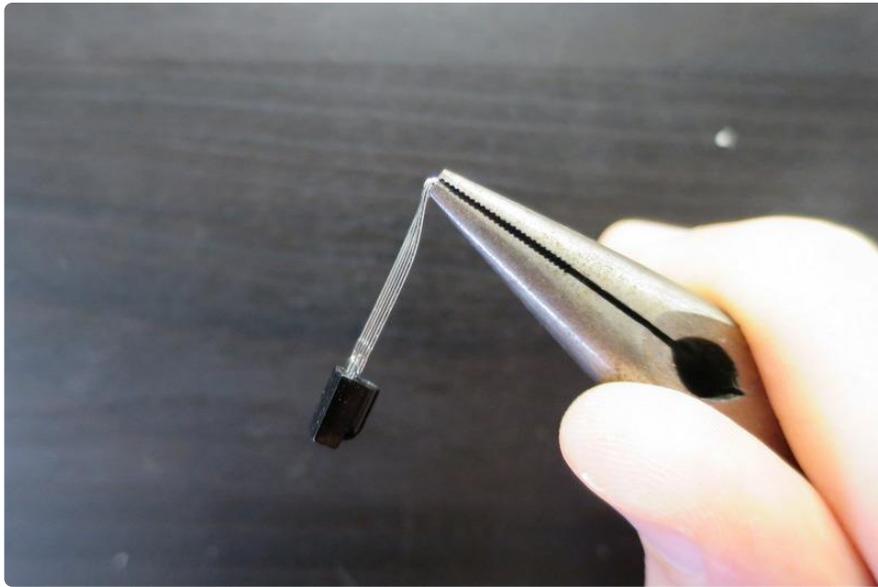
IR Receiver Cable

You can insert the three pins of the IR sensor into the the female-female Jumper Wires, but the connection will likely be a little loose.

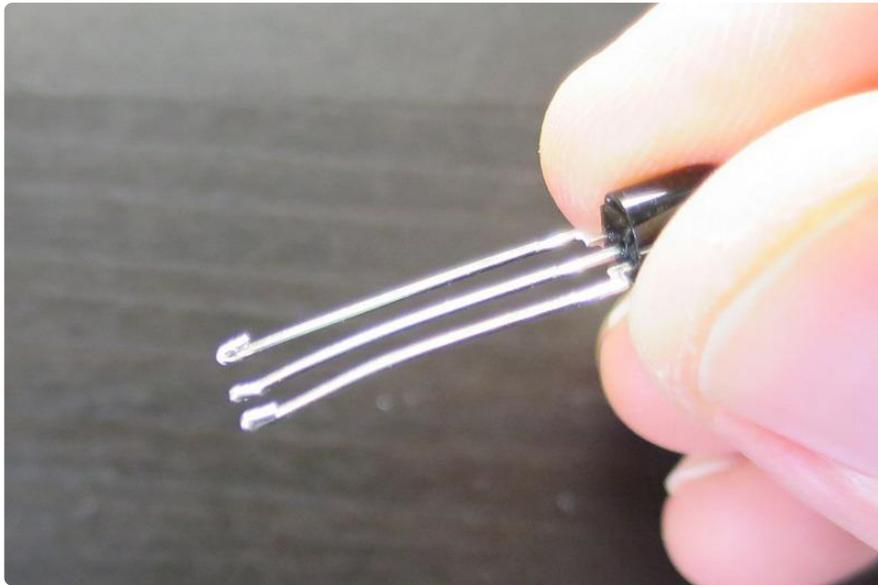


To make for a more secure connection, use needle nose pliers to bend the end of the sensor pins back on themselves to double up the thickness.

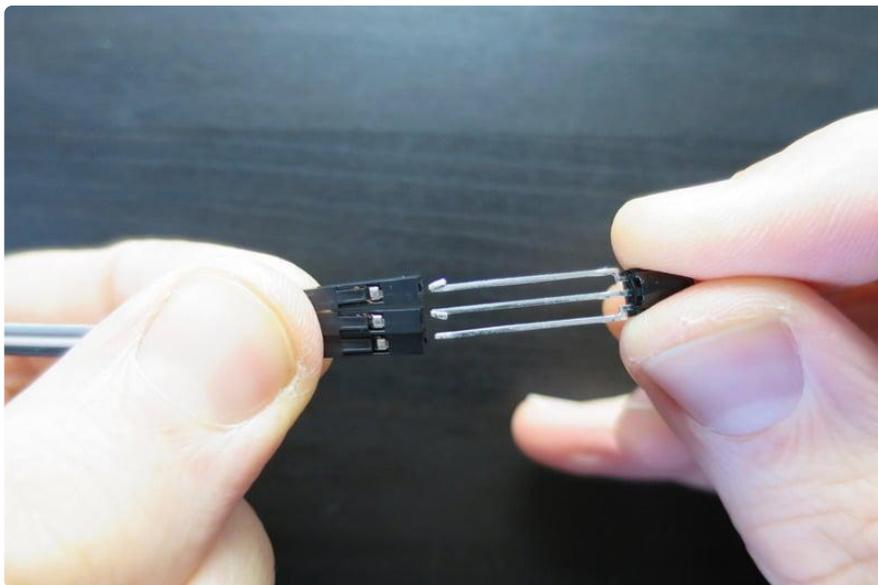




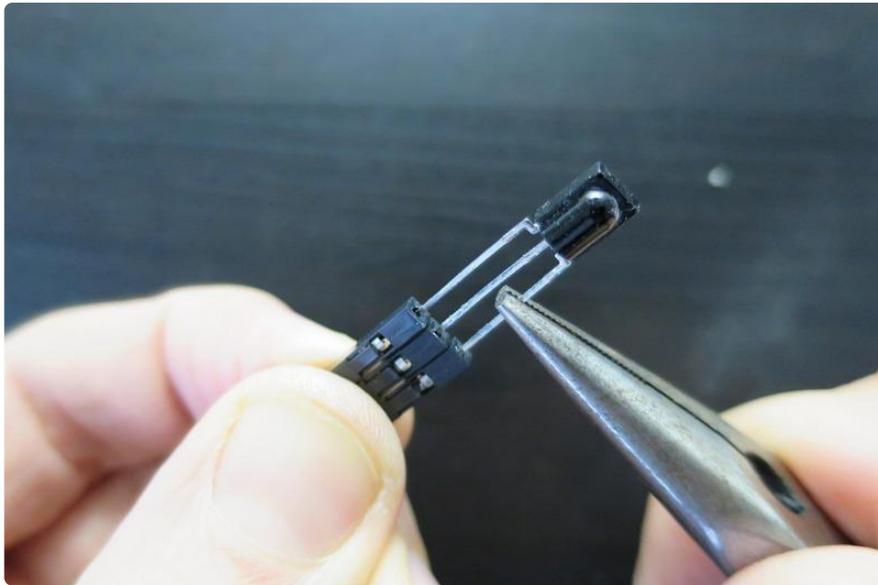
After the ends of the pins are bent back, use the pliers to flatten the ends out as much as possible.



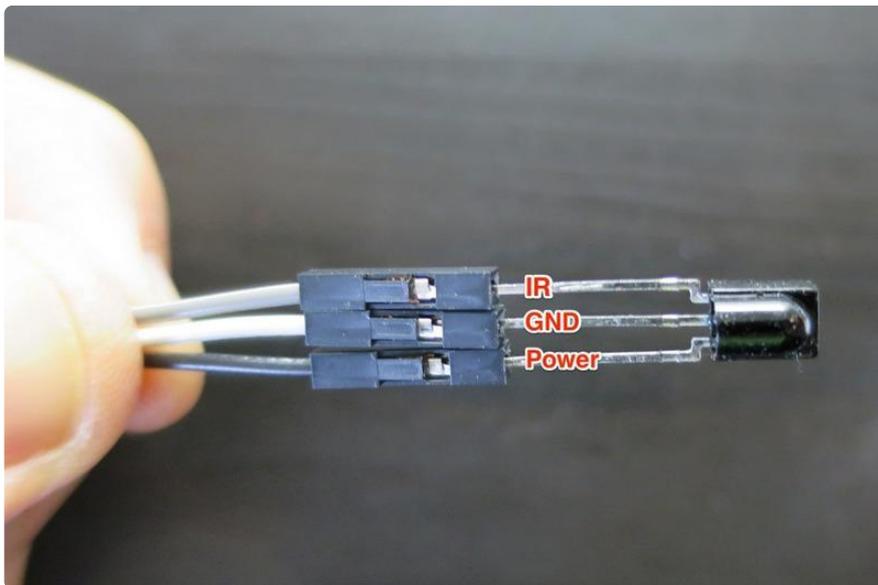
Now when you insert the pins into the jumper wires, the connection will be snug, and it may not be possible to pull the pins out again without damaging the jumper wires



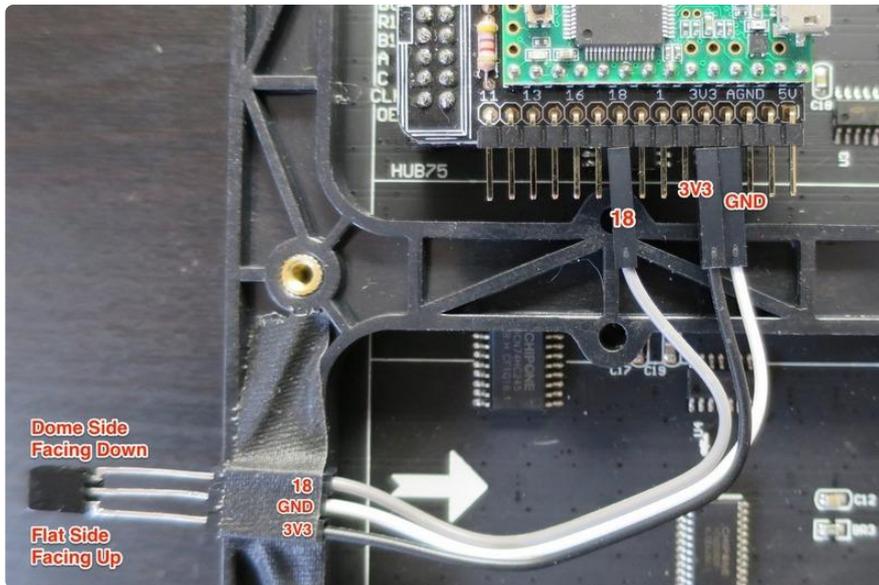
You may need to use the pliers to push the sensor pins into the crimps on the cable as it's now a tight fit.



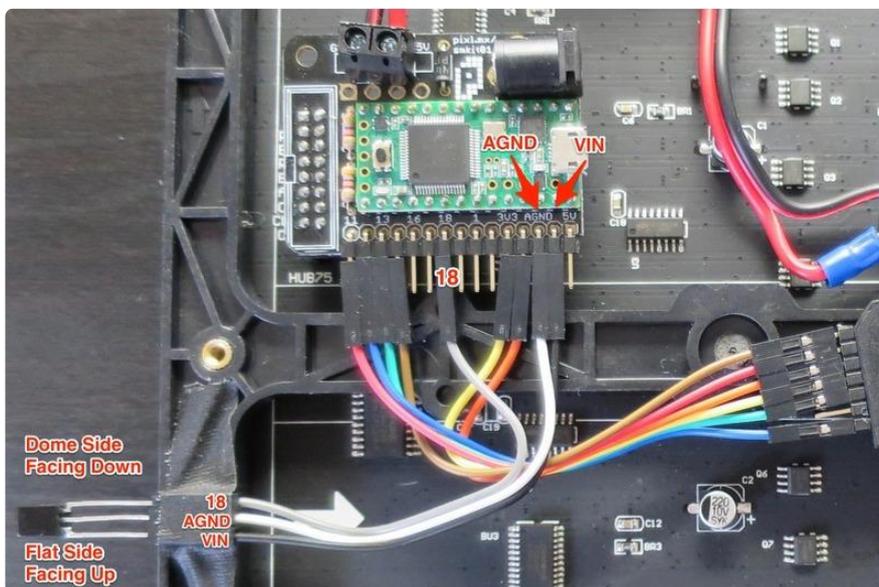
The sensor connects to power (3V3), ground, and the IR signal connects to an input pin (18). Use the rounded side of the sensor and this picture to identify the three pins.



Connect the sensor to the SmartMatrix Shield, and mount the sensor with the dome side facing out so it can receive signals from the IR remote. You can mount the sensor on the side of the panel or in another location it is likely to receive the infrared light from the remote. If you are mounting the panel inside an enclosure, you may want to use a longer cable to bring the IR sensor out of the enclosure. I used black gaffers tape to mount the sensor in the picture.



If you added a microSD reader following our earlier tutorial on the SmartMatrix Animated GIF Player, you will find the 3V3 and GND pins are already occupied. You could make custom cables to share the 3V3 and GND pins with two other connections, but that's a lot of extra work. Instead we will use the AGND pin in place of GND, and the VIN pin instead of 3V3. AGND connects to GND inside the Teensy 3.1 through an inductor. VIN will be between 4-5V instead of 3.3V, but both the Teensy 3.1 and IR sensor are 5V compatible, so there shouldn't be any problem running the sensor from VIN.



Programming Aurora Sketch

Lite Tool Installation

If you just want to run Aurora without changing any code, you can download a pre-compiled release version of Aurora and a simple software tool to load it onto the Teensy 3.1. If you want to be able to compile Aurora to add your own patterns or tweak how it works, skip this section and do the full install.

Compiled Aurora Sketch

Download the compiled version of the latest Aurora release [from GitHub \(\)](#). Extract the zip file.

There is an "sd" folder, which can be loaded onto a microSD card if you installed the microSD reader earlier. You can put the "aurora", "gifs", and "messages" folders into the root directory of the microSD card.

Find the Aurora.hex file in the "bin" folder. We will use this file in the next step.

Teensy Loader

To load a compiled sketch onto the Teensy 3.1 you need the Teensy Loader tool, which is available for Windows, OSX, and Linux [here \(\)](#).

Install following the instructions for your platform.

Once the Teensy Loader is installed and running, choose "Open Hex File" under the "File" menu, and locate the Aurora.hex file.

Connect the Teensy 3.1 via a USB cable to your computer, and power on the display. Press the button on the Teensy 3.1 to program the Aurora sketch.

You should now see a menu displayed on the LED panel, which can be controlled with the IR remote.

Full Installation

Installing the tools and libraries needed to compile Aurora can take some time. The instructions are too long to include here, but are available on the [Aurora Wiki \(\)](#).

Using Arduino Sketch

The video demo gives an overview of the features available in Aurora.

Remote Control Buttons



Global

- Play (top center) sleeps and wakes the display
- Vol-/Vol+ adjust brightness

Main Menu

- Navigate using the D-pad: Up/Down navigates through the menu, Left/Right changes the content preview
- Enter shows content or enters the Settings Menu

Animations

- Enter returns to the Main Menu
- "1" controls Autoplay modes
 - Pause - Stay on the current content until the left or right button is pressed.
 - Autoplay - Automatically switches to the next content after a short time.
 - Random - Automatically switches to a random content after a short time.
- "3" controls the message or clock overlays. Press to cycle through the clock overlay (only available if a RTC module is installed) and any messages stored on the SD Card. Hold to hide the current overlay.

Patterns

- Patterns uses the same buttons as Animations with one addition:
- "2" cycles through selections in the color palette on patterns that have a customizable palette

Customizing the Display

Customize the content on your display by loading GIFs and messages to the SD card. Follow the recommendations in the [SmartMatrix GIF Player guide \(\)](#) for finding and making your own GIFs. The messages feature is quite new and not documented yet, but you can use the examples as templates to create your own messages.

Troubleshooting

Nothing is displayed

- Make sure the sketch uploads successfully to the Teensy 3.1
- Check the soldering on the SmartMatrix Shield

"No SD card" message

- Make sure the microSD card is inserted in the reader, and is formatted properly
- Check your soldering on the expansion connector and Teensy
- Check the wires, make sure none are loose or connected to the wrong pins
- Try another microSD card

GIFs aren't playing

- See [Troubleshooting \(\)](#) in the SmartMatrix GIF Player section

Sketch does not respond to IR remote

- Make sure the battery still has charge
- Test that your remote is actually transmitting by [viewing the remote's LED through a digital camera \(\)](#)
- Check the wires, make sure none are loose or connected to the wrong pins
- If you are compiling the Aurora sketch yourself, make sure the CPU Speed is set to 48MHz or 96MHz (not 72MHz). You can first make sure the compiled Aurora.hex works on your display.
- If you're not sure if your IR sensor is working, try [wiring it up to an LED to test \(\)](#).

"No Clock" message

- The clock overlay requires a separate RTC module with one of these chips: DS1307, DS1337 and DS3231. Adafruit's [DS1307 breakout \(http://adafru.it/255\)](http://adafru.it/255) and the [ChronoDot \(http://adafru.it/264\)](http://adafru.it/264) should work with the sketch. The wiring is more complicated as you will have to share the GND and power pins. SDA connects to SmartMatrix Shield pin 17, SCL to pin 16. Make sure I2C pull-up resistors are installed either on the SmartMatrix Shield or RTC board.

Acknowledgements

[Jason Coon \(\)](#) - Designing the Aurora sketch was a team effort, but Jason did all of the coding.

[Craig Lindley \(\)](#) - Original author of the GIF decoder used in Aurora, and his [Light Appliance project \(\)](#) was the initial inspiration for making Aurora.

[Daniel Garcia and Mark Kriegsman's FastLED Library \(\)](#) - Most of the patterns in Aurora use FastLED and many of them are examples contributed to the [FastLED Community \(\)](#). The FastLED Library supports many types of individual pixels and LED strips, not just matrix displays and is worth checking out for any LED project.

[Discord Games \(\)](#) - The Pixel Art used in the demo is used with permission from Discord Games, Inc., makers of [Chasm \(\)](#)

[Blender Foundation \(\)](#) - The animated video used in the demo is (c) copyright 2008, Blender Foundation

Guide written by Louis Beaudoin (Pixelmatix.com) CC-BY-SA