Adafruit High Power Infrared LED Emitter

Created by Liz Clark

https://learn.adafruit.com/adafruit-high-power-infrared-led-emitter

Last updated on 2023-08-29 04:52:24 PM EDT
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td>Pinouts</td>
<td>5</td>
</tr>
<tr>
<td>• Power Pins</td>
<td></td>
</tr>
<tr>
<td>• Input Pin</td>
<td></td>
</tr>
<tr>
<td>• STEMMA JST PH</td>
<td></td>
</tr>
<tr>
<td>• Optional 5mm LED</td>
<td></td>
</tr>
<tr>
<td>• Signal LED and LED Jumper</td>
<td></td>
</tr>
<tr>
<td>• Power LED and LED Jumper</td>
<td></td>
</tr>
<tr>
<td>CircuitPython</td>
<td>6</td>
</tr>
<tr>
<td>• CircuitPython Microcontroller Wiring</td>
<td></td>
</tr>
<tr>
<td>• CircuitPython Usage</td>
<td></td>
</tr>
<tr>
<td>• Example Code</td>
<td></td>
</tr>
<tr>
<td>Python Docs</td>
<td>8</td>
</tr>
<tr>
<td>Arduino</td>
<td>9</td>
</tr>
<tr>
<td>• Other Arduino Code Options</td>
<td></td>
</tr>
<tr>
<td>• Wiring</td>
<td></td>
</tr>
<tr>
<td>• Library Installation</td>
<td></td>
</tr>
<tr>
<td>• Example Code</td>
<td></td>
</tr>
<tr>
<td>Arduino Docs</td>
<td>12</td>
</tr>
<tr>
<td>Downloads</td>
<td>12</td>
</tr>
<tr>
<td>• Files</td>
<td></td>
</tr>
<tr>
<td>• Schematic and Fab Print</td>
<td></td>
</tr>
</tbody>
</table>
*Pew* *pew*! This board is like a little ray gun for infrared light, with two high powered LED outputs. When controlled with the onboard N-Channel FET driver, you'll be blasting 100mA-200mA of current pulsing through each LED for 10+ meters of range! This is the easiest way to get great IR emitter range without wiring up a bunch of parts.

By using a 2mm pitch STEMMA JST PH cable () with headers or alligator clips on the end, you can easily wire this board up without any soldering at all.
This board features a power transistor so you just need to provide about 3-5V DC power to the V+ power in, ground for ground, and then a 3-5V logic level signal on the input pin. When the pin is high, the LEDs are on, when the signal pin is low, the LEDs are off. Since humans don't see in IR, there's a red LED that will let you know when the IR LEDs are lit. Two LEDs are pre-soldered to the PCB, one pointing up and one pointing out. If you want even MORE coverage, there's a spot to solder in a separate 5mm IR LED (not included).

If powering with 5V, the board will draw about 200mA per LED (400mA total) when pulsing on. If powering with 3V the board will draw about 100mA per LED (200mA total). We don't recommend keeping the LEDs on for long periods at this current draw, they're for remote control transmission more than IR illumination.

Each STEMMA board comes with a fully assembled and tested PCB but no cable. No soldering is required to use it, but you will need to pick up a 2mm pitch, 3-pin STEMMA JST PH cable. Alternatively, if you do want to solder, there's a 0.1" spaced header for power / ground / signal.
Pinouts

Power Pins

- V+ - this is the power pin. We have included an N-Channel FET driver on board for the IR LEDs that will take 3-5VDC. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V microcontroller like Arduino, use 5V.
  - If powering with 5V, the board will draw about 200mA per LED (400mA total) when pulsing on.
  - If powering with 3V the board will draw about 100mA per LED (200mA total).
- GND - common ground for power and logic.

Input Pin

- In - this is the signal input pin. When the pin is high, the IR LEDs are on, when the signal pin is low, the IR LEDs are off.

STEMMA JST PH

- STEMMA JST PH () - 2mm pitch STEMMA JST port with connections for V+, GND and In. Plug in a 3-pin STEMMA JST PH cable () with headers or alligator clips on the end and you can easily wire this board up without any soldering at all.

Optional 5mm LED

- On the back of the board, there is a spot to solder an optional 5mm IR LED (). This can be used if you want additional IR coverage.
Signal LED and LED Jumper

- Signal LED - In the lower left corner, below the STEMMA connector, on the front of the board, is the signal LED, labeled Sig. It is the red LED. Since humans don't see in IR, the Signal LED lights up to let you know when the IR LEDs are lit.
- Signal LED jumper - This jumper is located on the back of the board and is labeled Sig. Cut the trace on this jumper to cut power to the "Sig" LED.

Power LED and LED Jumper

- Power LED - In the upper left corner, above the STEMMA connector, on the front of the board, is the power LED, labeled on. It is the green LED.
- Power LED jumper - This jumper is located on the back of the board and is labeled On. Cut the trace on this jumper to cut power to the "on" LED.

CircuitPython

It's easy to use the High Power Infrared LED Emitter with Python or CircuitPython, and the Adafruit_CircuitPython_IRRemote() module. This module allows you to easily write Python code that allows you to send IR remote pulses using the pulseio() core module.

CircuitPython Microcontroller Wiring

First, wire up an infrared LED emitter to your board exactly as shown below. Here's an example of wiring a Feather M4 to the infrared LED emitter using one of the handy STEMMA JST PH() cables:

Board 3V to LED emitter V+ (red wire)
Board GND to LED emitter GND (black wire)
Board pin 5 to LED emitter In (white wire)

You can also use standard 0.100" pitch headers to wire it up on a breadboard:
CircuitPython Usage

To use with CircuitPython, you need to first install the IRRemote library, and its dependencies, into the lib folder on your CIRCUITPY drive. Then you need to update code.py with the example script.

Thankfully, we can do this in one go. In the example below, click the Download Project Bundle button below to download the necessary libraries and the code.py file in a zip file. Extract the contents of the zip file, and copy the entire lib folder and the code.py file to your CIRCUITPY drive.

Your CIRCUITPY/lib folder should contain the following folders and file:

- adafruit_irremote.mpy

```python
# SPDX-FileCopyrightText: 2022 Liz Clark for Adafruit Industries  
# SPDX-License-Identifier: MIT  
# Based on irremote_transmit.py for CPX by ladyada

import time  
import pulseio  
import board
```

© Adafruit Industries
import adafruit_irremote

# Create a 'PulseOut' to send infrared signals on the IR transmitter @ 38KHz
pulseout = pulseio.PulseOut(board.D5, frequency=38000, duty_cycle=2**15)
# Create an encoder that will take numbers and turn them into NEC IR pulses
emitter = adafruit_irremote.GenericTransmit(
    header=[9500, 4500], one=[550, 550], zero=[550, 1700], trail=0
)

# count variable
count = 0
while True:
    # send IR pulse
    emitter.transmit(pulseout, [255, 2, 255, 0])
    # increase count
    count += 1
    # print to REPL
    print("IR signal sent %d times!" % count)
    # two second delay
    time.sleep(2)

Once everything is saved to the CIRCUITPY drive, connect to the serial console () to see the data printed out!

```
soft reboot

code.py output:
IR signal sent 1 times!
IR signal sent 2 times!
IR signal sent 3 times!
IR signal sent 4 times!
IR signal sent 5 times!
IR signal sent 6 times!
IR signal sent 7 times!
IR signal sent 8 times!
IR signal sent 9 times!
```

The code begins by creating a pulseio object on pin D5. This will send infrared signals via the IR transmitter at 38KHz. Then, emitter is created with the adafruit_irremote library that will take numbers and turn them into NEC IR pulses.

In the loop, an IR signal is pulsed out every two seconds. In the REPL, you'll see "IR signal sent # times!" print out after the pulse is sent. The variable count counts the number of times this occurs and is passed to the string in the REPL.

---

**Python Docs**

[Python Docs ()](https://docs.python.org/3/)
Arduino

Using the High Power Infrared LED Emitter with Arduino involves wiring up the infrared LED emitter to your Arduino-compatible microcontroller, installing the IRremote library and running the provided example code.

Other Arduino Code Options

There are many libraries available for sending and decoding infrared signals with Arduino. The Using an Infrared Library on Arduino Learn Guide by Chris Young that goes into more detail on other libraries available and their usage. You can also send IR signals without a library, as shown in the original IR Sensor Learn Guide.

Wiring

Wire as shown for a 5V board like an Uno. If you are using a 3V board, like an Adafruit Feather, wire the board's 3V pin to the infrared LED emitter V+.

Here is an Adafruit Metro wired up to the infrared LED emitter using the STEMMA JST PH cable:

- Board 5V to LED emitter V+ (red wire)
- Board GND to LED emitter GND (black wire)
- Board pin 3 to LED emitter In (white wire)

Here is an Adafruit Metro wired up using a solderless breadboard:
Library Installation

You can install the IRremote library for Arduino using the Library Manager in the Arduino IDE.

Click the Manage Libraries ... menu item, search for IRremote, and select the IRremote library:
Example Code

To launch the example code in the Arduino IDE, click on File - Examples - IRremote and select SimpleSender. The example code can also be found on GitHub in the library repository.

```cpp
void loop() {
  /* Print current send values */
  Serial.println();
  Serial.println("Send new address=0x00");
  Serial.println(Address, HEX);
  Serial.println("Send command=0x00");
  Serial.println(Command, HEX);
  Serial.println("Send repeted=0x00");
  Serial.println(RepeatCount);
  Serial.println();
  Serial.print("Send NEC with 16 bit address");
  Serial.flush();

  // Receive output for the first loop must be: Protocol=IR Address=0x00 Command=0x04 Raw Data=0x030402  (16 bits)
  IrSender.sendNEC(Address, xCommand, xRepeatCount);

  if (xRepeatCount) {
    Serial.println("Send NEC wth 0x030402");
    IrSender.sendNEC(xAddress, xCommand, xRepeatCount);
  } else {
    Serial.println("Send NEC address 0x030402");
  }

  xAddress = 0x0001;
  xCommand = 0x01;
  xRepeatCount = 0;
```

©Adafruit Industries
Upload the sketch to your board and open up the Serial Monitor (Tools -> Serial Monitor) at 115200 baud. You should see your library version and the pin number that will be sending the IR pulses print to the Serial Monitor. Then, the `address`, `command` and `repeats` are printed, followed by the message **Send NEC with 16 bit address**. The `address` and `command` will incrementally increase as the code runs.

---

**Arduino Docs**

[Arduino Docs](#)

**Downloads**

**Files**

- Infrared LED Datasheet
- EagleCAD PCB files on GitHub
- 3D models on GitHub
- Fritzing object in the Adafruit Fritzing Library
Schematic and Fab Print