## HEG Open Hardware Designs

MIT License
Copyright 2019, Joshua Brewster

Permission is hereby granted, free of charge, to any person obtaining a copy of this hardware, software, and associated documentation files (the "Product"), to deal in the Product without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Product, and to permit persons to whom the Product is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Product.

THE PRODUCT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE PRODUCT OR THE USE OR OTHER DEALINGS IN THE PRODUCT.

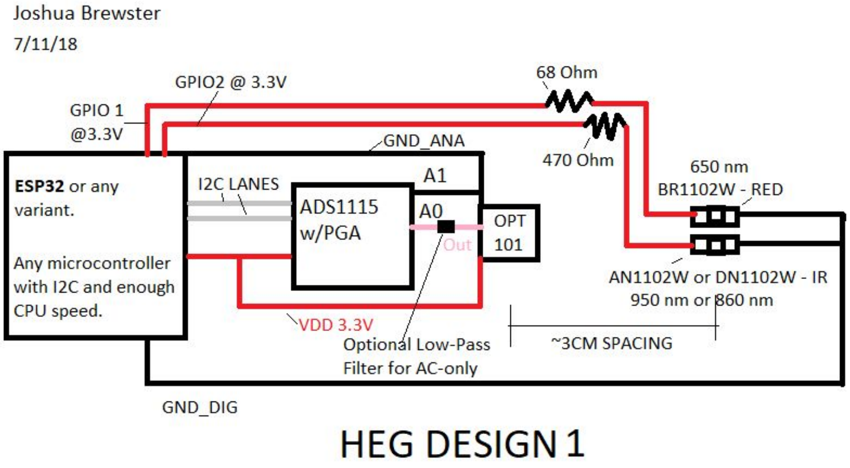
### HEGduino kit

This design is based off the original, now-expired HEG patent 5995857 by Hershel Toomim and Bob Marsh. The original sensor design is simply an OPT101 spaced at 3cm from 2 50mA LEDs - one in the Red spectrum and one in the Infrared spectrum. The Red LED is set at 650nm and the Infrared LED is set at either 850nm or 950nm. 850nm provides more linear scaling for blood flow ratio, while 950nm is the industry standard for pulse oximetry.

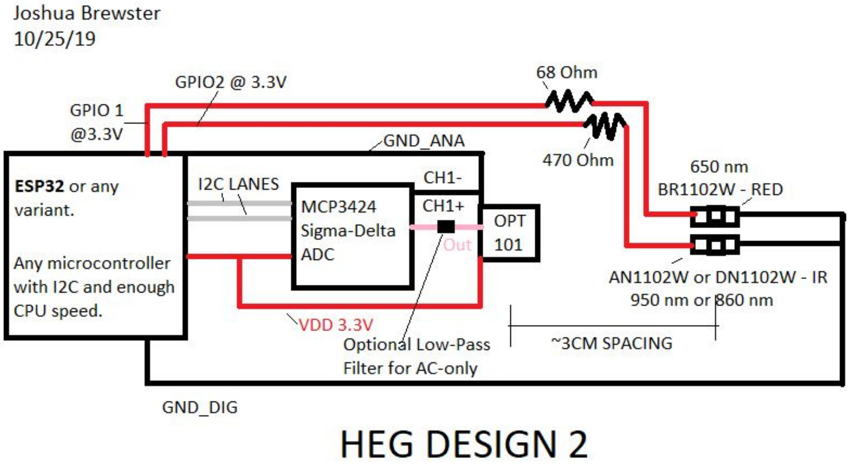
Our microcontroller of choice is the Espressif ESP32 using Arduino libraries, available for ~5 dollars on average, with our firmware available for free under the MIT license at https://github.com/moothyknight/HEG_ESP32. We also originally accomplished it with an Arduino Nano v3 which we purchased for 1 dollar apiece plus shipping. We are utilizing the ESP32 for its IoT capabilities which allows full interaction via a local webserver on the device itself. It also can connect online for remote data collection abilities or use bluetooth as well as traditional serial methods. None of our work is original on this in the sense that we used wholly open source libraries to develop the software solution, aside from our own simple method for driving the LEDs and timing the readings.

Continue below for multiple designs.

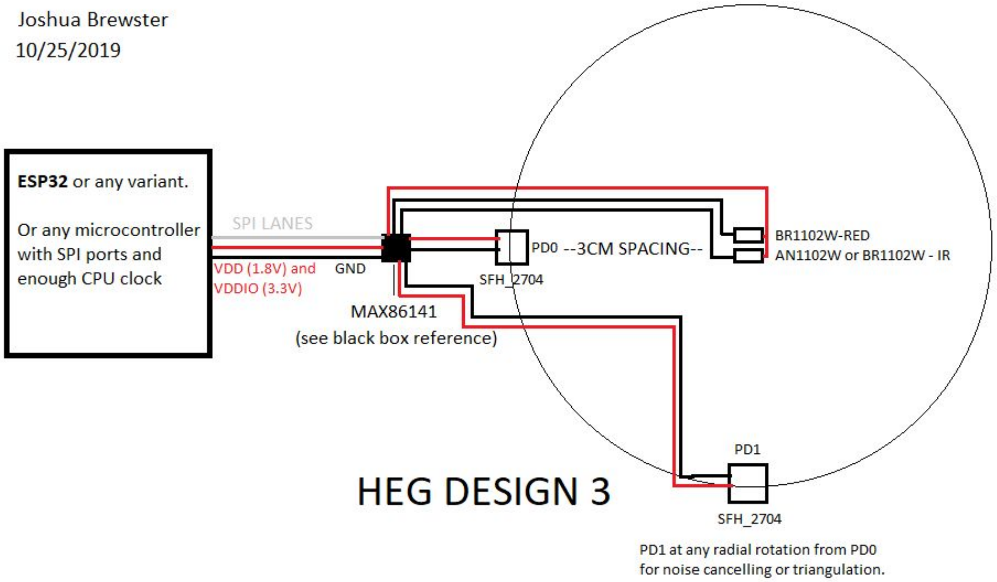
**Varied designs (thereby classifying them under the MIT License for open source use):**



Design 1 as described above, in simplified form.



Design 2 with MCP3424, not much different with better Sigma-Delta and resolution. Drivers coming soon.



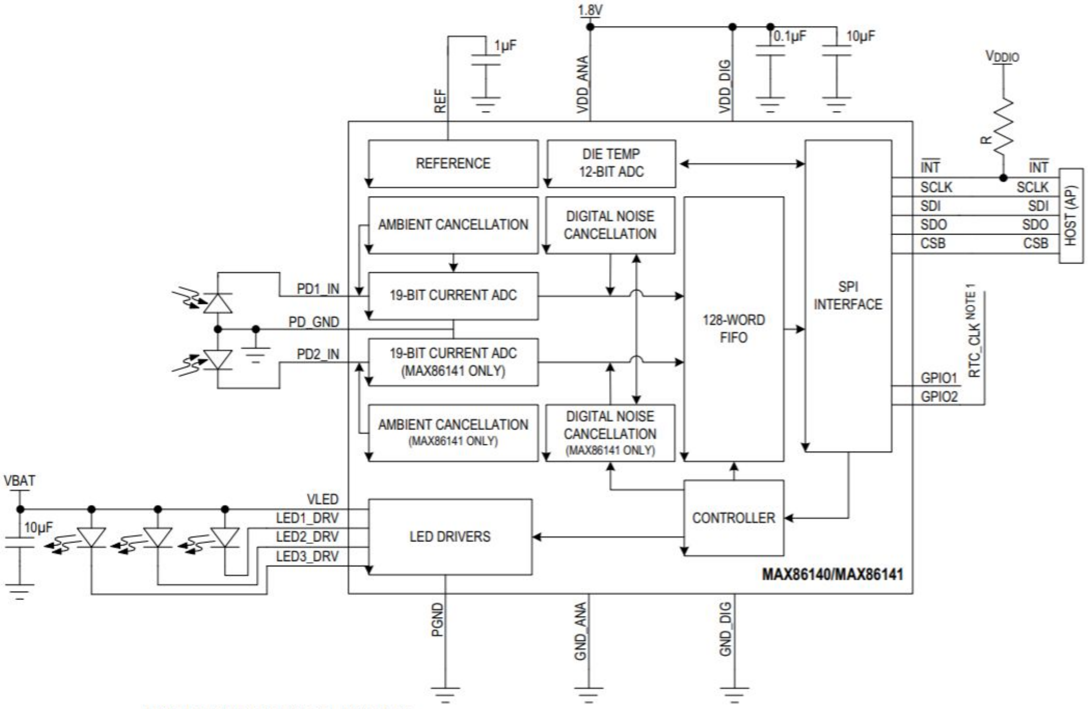
Joshua Brewster
10/25/2019

**ESP32** or any variant.
Or any microcontroller with SPI ports and enough CPU clock
SPI LANES
VDD (1.8V) and VDDIO (3.3V)
GND
MAX86141 (see black box reference)
PD0 --3CM SPACING--
SFH_2704
BR1102W-RED
AN1102W or BR1102W - IR
PD1
SFH_2704

HEG DESIGN 3

PD1 at any radial rotation from PD0
for noise cancelling or triangulation.

Experimental MAX86141 design. The wiring follows the 2-sensor recommendations in the datasheet by MAXIM to the T (see below), with our novel sensor spacing and these photodiodes. This allows for 4200 samples per photodiode per second at 18-bit resolution, opening up the possibility of Fast Optical Signal methodologies and at a highly affordable rate. Drivers coming soon.



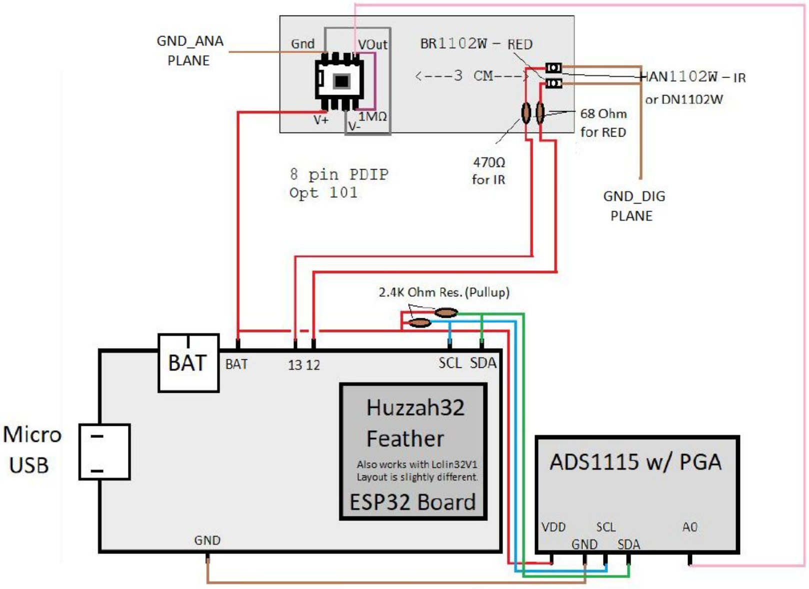
NOTE 1: WHEN GPIO_CTRL[3:0] = 0001 ONLY.

Abov is the reference sheet provided by MAXIM, we followed this exactly and included ground planes wherever possible for improved isolation and decoupling. The MAX86141 should be as close to the photodiode site as possible.

As you can see, it’s fairly easy to design these things, the key is finding the best and most affordable parts - which improve year to year. These are a solid foundation to begin from. And yes, I used Paint for these diagrams… but I recommend EAGLE or KiCAD for some of the better PCB design tools.

More design notes below...

More detailed view of Design 1, minus the optional GND_ANA to A1 on the ADS1115 (for differential output):



Design notes:

There are several key design ideas to keep in mind when developing the sensor.
1st: The analog and digital components need separate ground lines/planes. The analog output should also be shielded with copper planes to minimize noise and act as a low pass filter.

2nd: The closer you can get the ADC to the signal output the better, movement can generate static and add noise if there is more wire between the ADC and sensor.

3rd: You don't need more than 100 samples per second and 16 bits for good signal fidelity. We demonstrated this with a minimum-bar ADS1115, which includes PGA and differential modes to get pretty decent signal. We also recommend the MCP3424, an 18-bit, 240sps sigma-delta ADC which is more than enough for pulse oximetry, and simultaneous sensor data. These are both available as arduino kit-styled components.