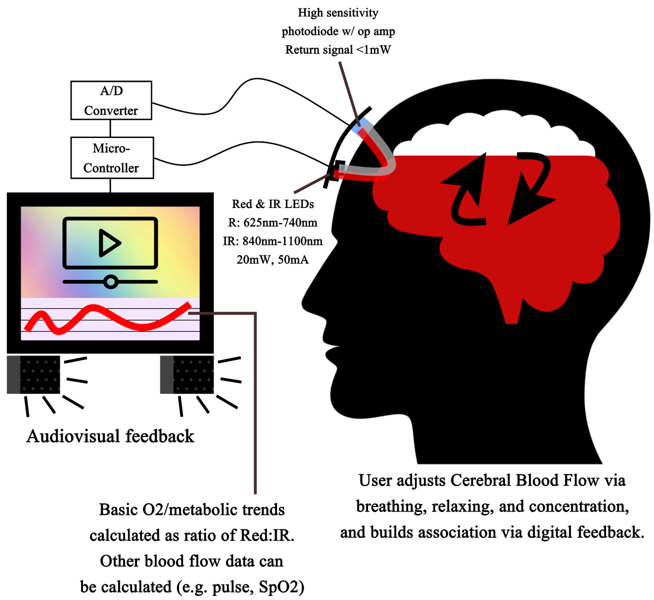
Ultra Low Cost FNIRS headband design, based on HEGduino (free source)
by Josh Brewster



Overview:

This project involves developing a low-cost Functional Near-Infrared Spectroscopy (FNIRS) headband, based on the free-source HEGduino platform. The hardware is essentially a pulse oximeter designed with specific sensor spacing to resolve deeper tissue signals. This technology, known as Hemoencephalography (HEG) in biofeedback applications, monitors blood flow and provides functional imaging for brain blood flow studies. We aim to create an affordable device for stress and breathing training, functioning similarly to an HRV monitor but worn on the head. I’ve included two possible specifications, we should quote for both as the first should be a little more straightforward but the second one is more scientifically valid and medically relevant.

Design 1 Hardware Specifications (lower fidelity than Design 2):

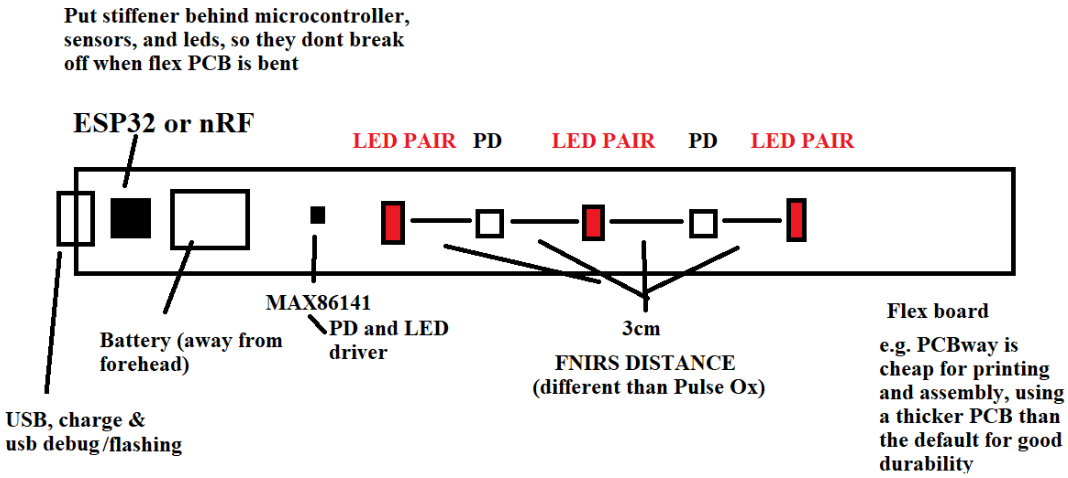
- **Microcontroller:**
  - **ESP32 WROOM32**: Alternatively, the smaller ESP32-S (Bluetooth only) can be used. Downclock to 80MHz to extend battery life.
  - **nRF52810 or nRF52832**: E.g., Fanstel BT832A or BT832, which offer better power efficiency. Both can be bootstrapped quickly with platform.io or Arduino and ready-made sensor libraries.
- **Pulse Oximeter Sensor:**
  - **MAX86141**: Dual-channel sensor, capable of driving up to 6 LEDs.
  - **Photodiodes**: BPW-34S, these are the best available and low cost.
  - **LEDs**:
    - **Domed 660nm or 740-770nm Red/Far Red LED**: Domed LEDs preferred for a narrower cone of light created. Power level: 20mAh-50mAh (e.g., XZM2MR55W-3 by SunLED). Far red is better, just the pickings are slimmer.
    - **Domed 840-880nm Infrared LED**: Same power output level (e.g., XZTHI55W-3 by SunLED).
- **Power and Connectivity:**
  - **USB type C**: For battery charging and debugging. Include TVS diodes on the battery charge portion for safety compliance.
  - **Lipo battery**, safe, cheap, any size.

Firmware Requirements:

- **Sensor Data Acquisition**:
  - **MAX86141**: Simultaneously senses photodiodes, filtering, and processing the signals internally.
  - **LED Driving**: LEDs should operate at the highest rate possible, aiming for 50Hz-100Hz updates for optimal accuracy. Josh can advise on firmware settings.
- **Data Transmission**:
  - **Bluetooth**: The microcontroller should transmit buffered sensor data byte packets. Data should be packed efficiently (e.g., 24-bit structure for 19-bit MAX86141 values).
- **Commands and Debugging**:
  - **Bluetooth Commands**: Simple commands to power the device on/off and an inactivity timer for automatic shutdown.
  - **USB Debug Output**: Output the same sensor packet data to USB as Bluetooth for consistency in parsing on the client side.

Visual Diagram

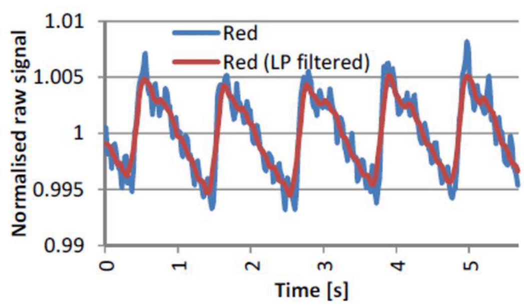
Idea is a flex PCB that covers the forehead with a row of sensors and LEDs:



The diagram illustrates a flexible PCB covering the forehead with a row of sensors and LEDs. The design ensures that LED light does not leak to the photodiodes, requiring contact with the skin for accurate readings. Neoprene or similar materials may be used for comfort and to ensure a good seal.

For designing a case, the LEDs cannot leak light to the Photodiodes, all light must be pass via contact with the skin (forehead). The Mendi protrudes the LED and Photodiode contacts out, alternatively something soft that can conform to skin would be better and more comfortable, we just used neoprene before but we need a good seal on the sensors.

Output Expectations:

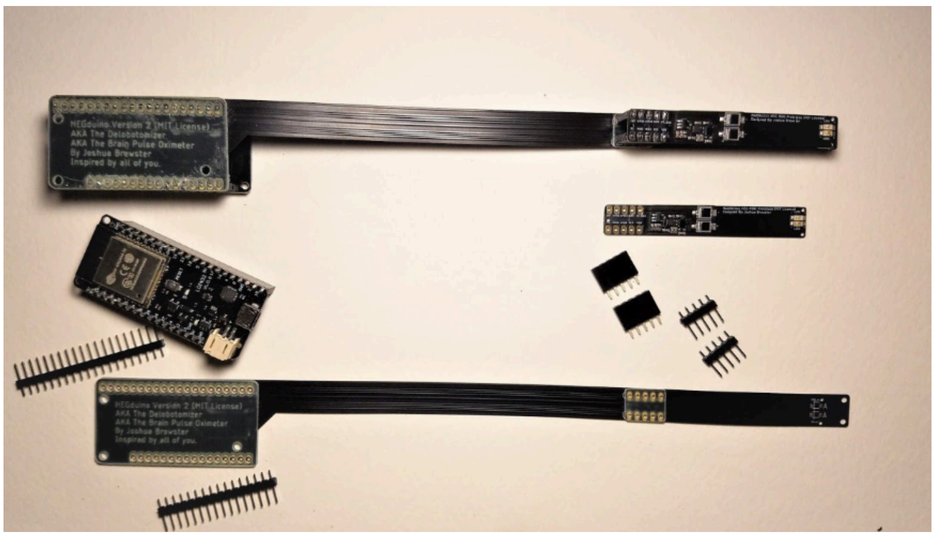


The expected data resembles typical pulse oximetry readings but with signals indicative of blood flow changes in deeper tissues. We'll have one of these lines for each LED, red and infrared are compared to infer blood oxygen changes, and with multiple LED pairs we can compare across multiple sites.

Old Resources: https://github.com/joshbrew/HEG_ESP32_Delobotomizer

FNIRS sensor BOM:
https://docs.google.com/spreadsheets/d/1MgO6r2GJB0lHYTsHTYTxhSiRR5hWRgtUIwzvLXI5KvM/edit?usp=drive_link

MAX86141 BOM:
https://docs.google.com/spreadsheets/d/1BV-0n5jjRR3LmGyLHRKKsdT_74nFn3pC/edit?usp=drive_link&ouid=115019542711496588778&rtpof=true&sd=true



This configuration put both PDs together but we are looking for a different layout this time that gives us the fuller breadth of the forehead.

Design 2, superior hardware specs, state-of-the-art for this price point:

Hardware:

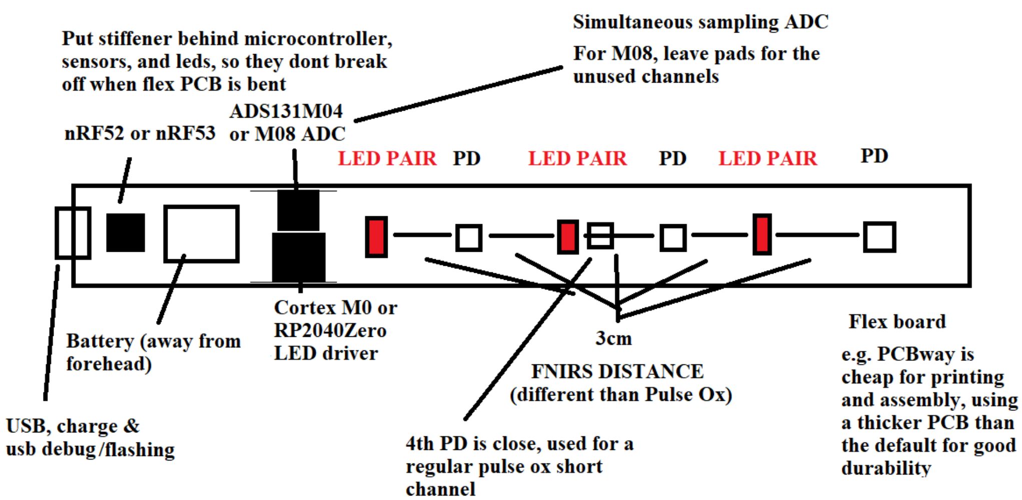
- **Microcontroller**:
  - **nRF52 or 53**: For BLE5.
  - **Cortex M0 or RP2040Zero**: Independent LED driver with up to 20 PWMs, requiring synchronization of ADS131 readings with LED pulses. RP2040 Zero may need a Darlington Transistor to increase current yield but we don't need more than 20mAh, the RP2040 should be able to do up to 50mAh.
- **Sensor ADC**:
  - **ADS131M08 or M04**: 4-8 channels 24-bit sigma delta, simultaneous sampling ADC.
  - **Photodiodes**: BPW-34S, these are the best available and low cost.
  - **LEDs**:

        - **Domed 660nm or 740-770nm Red/Far Red LED**: Domed LEDs preferred for a narrower cone of light created. Power level: 20mAh-50mAh (e.g., XZM2MR55W-3 by SunLED). Far red is better, just the pickings are slimmer.
        - **Domed 840-880nm Infrared LED**: Same power output level (e.g., XZTHI55W-3 by SunLED).
- **Power and Connectivity**:
    - **USB type C**: For battery charging and debugging. Include TVS diodes on the battery charge portion for safety compliance.
    - **Lipo battery**, safe, cheap, any size

Firmware Notes:

- Special care is required to synchronize PWM signals with ADS131M08 for precise LED pulse capturing.
- The overall functionality remains similar to the simpler design but with enhanced signal resolution.

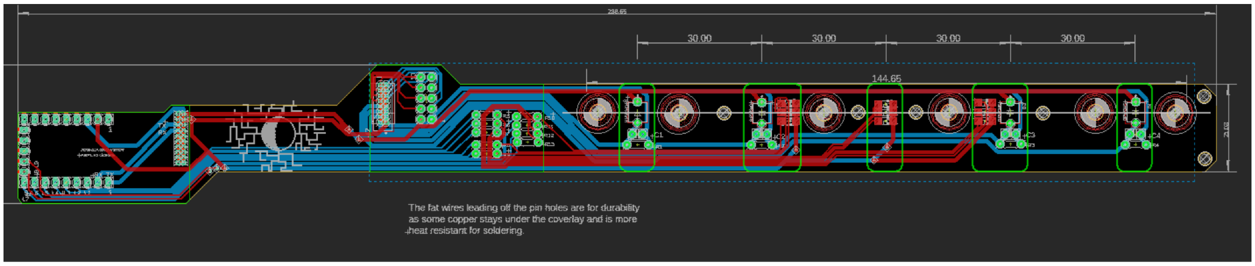
More sophisticated design, same relative cost as the MAX86141 but much bigger punch:

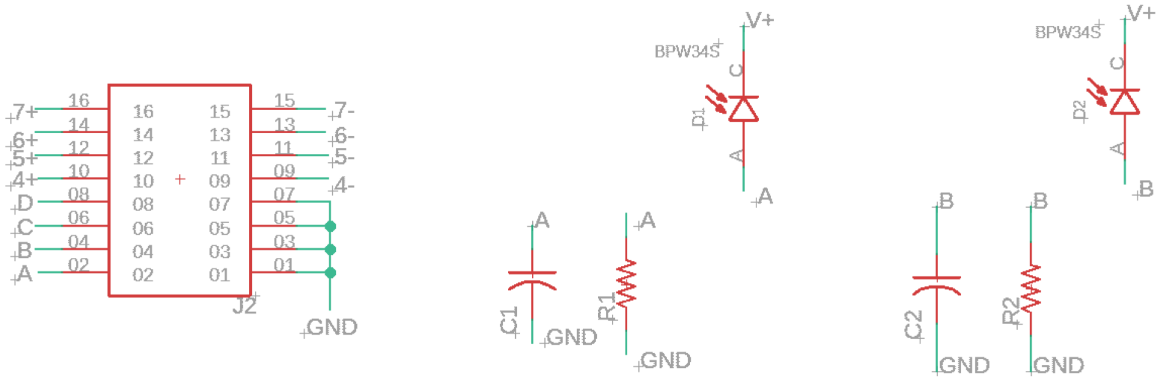


Note the short channel here is something seen on more advanced FNIRS designs for noise cancellation. It's the best configuration we could come up with for 4 channels as it may get too wide otherwise. Based on the pairings this gives 5 FNIRS sites and maybe more if it's sensitive enough to pick up the farther distances. We are working on testing this but the design is sound.Also for the non sensing components, keep those backside to not obstruct the forehead or bluetooth chip, power, etc.

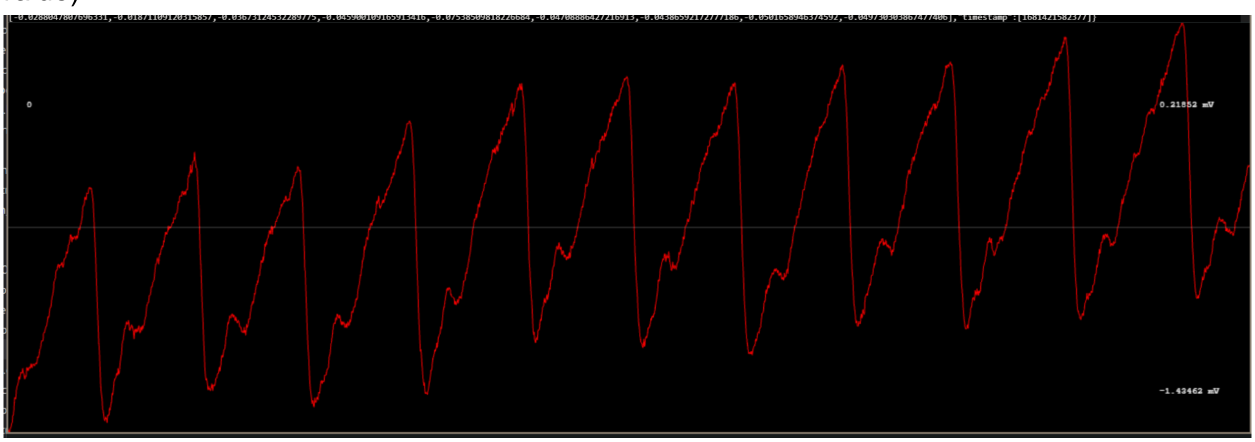
Sample Designs and Additional Resources:

Our sample designs include development boards for RP2040/Cortex M0 and an nRF + ADS131M08 setup. The signal quality achieved is state-of-the-art.





The photodiodes are configured in a simple low pass configuration, with negative ADC inputs to ground in this sigma-delta configuration. 1MOhm R and 220pF C (need to double check cap value)



This is raw PPG data we got off this specification through our finger. This is ultra high quality data.

nRF Resources with an FDA-approvable PCB design:
https://github.com/joshbrew/nRF5x-Biosensing-Boards

FNIRS sensor BOM:
https://docs.google.com/spreadsheets/d/1MgO6r2GJB0lHYTsHTYTxhSiRR5hWRgtUIwzvLXI5KvM/edit?usp=drive_link

BOM samples:
E.g BT840
https://docs.google.com/spreadsheets/d/1_HJGfFERPbjsfyVsvAEBeEDxsZvMkjib/edit?usp=sharing&ouid=115019542711496588778&rtpof=true&sd=true

E.g. BT40
https://docs.google.com/spreadsheets/d/1iG1TCaeJnqPSEQdyyh46KBVxkoCE-ekg/edit?usp=sharing&ouid=115019542711496588778&rtpof=true&sd=true

Another BT40 BOM (high overages though, not that necessary)
https://docs.google.com/spreadsheets/d/1nfP5cUY5rJFIu8GrZoNSI9P1zyI8ZTMB/edit?usp=sharing&ouid=115019542711496588778&rtpof=true&sd=true
E.g BT832 cheap nRF52 option (note BT832A is the nRF52810, very small memory)
https://docs.google.com/spreadsheets/d/1Iq_YZKACqgZNz5KGheo3KygyKR9DUOsgQVpr5Snjju0/edit?usp=sharing

https://docs.google.com/spreadsheets/d/1RUImk_B_wu3TRTACJXpPSCaapPaiVIjje_XyH6pws34/edit?usp=drive_link Peripheral stuff (not all directly relevant, just dev board sourcing, potential cheap route for sensor prefabs)